

La logica come strumento dell'intelligenza

Crisi fondazionale, collasso simbolico e matematica dell'intelligenza artificiale

Nota al lettore: struttura e logica dei tre livelli

Questa monografia è organizzata su tre livelli distinti, ciascuno con funzione propria. Conoscere la logica di questa struttura consente di orientarsi nella lettura e di scegliere il percorso più adatto ai propri interessi.

Il **primo livello** è costituito dalle **Parti I–V**: il corpo argomentativo principale. Qui si svolge l'argomento centrale — che la logica è uno strumento prodotto dall'intelligenza, non il suo fondamento — attraverso cinque movimenti successivi: la crisi fondazionale del programma hilbertiano (I), il collasso dell'AI simbolica (II), la matematica dei sistemi neurali (III), la biologia evolutiva della cognizione (IV), la ricomposizione teorica e la tesi finale (V). Questo livello è autosufficiente: un lettore che percorra soltanto le Parti I–V riceverà l'argomento nella sua interezza.

Il **secondo livello** è costituito dalle **Espansioni A–F**: approfondimenti tematici che ampliano e contestualizzano aspetti specifici dell'argomento principale. Non tutte le espansioni hanno lo stesso rapporto con la tesi centrale. Le Espansioni A (complessità algoritmica), B (PAC learning e generalizzazione) ed E (P vs NP e complessità computazionale) *rafforzano* strutturalmente la tesi: i loro risultati sono necessari per rispondere alle

obiezioni tecniche più serie. Le Espansioni C (co-evoluzione gene-cultura), D (filosofia della matematica pura) ed F (struttura temporale della cognizione) la *contestualizzano* in senso più ampio, inserendola in quadri disciplinari adiacenti. Il lettore non specialista può leggere A, B, E e rimandare C, D, F a un secondo momento senza perdere il filo dell'argomentazione.

Il **terzo livello** è costituito dalle **Appendici Tecniche I–VI**: formalizzazioni matematiche complete destinate ai lettori con formazione specialistica in logica, teoria dell'informazione, geometria differenziale e teoria delle categorie. Le Appendici I (PAC learning e teoria VC), II (geometria dell'informazione e gradiente naturale) e III (ottimizzazione variabile) sono strettamente connesse all'argomento principale. Le Appendici IV (teoria delle categorie applicata all'AI), V (complessità di Kolmogorov) e VI (teoria degli ordini e reticoli) contengono risultati di notevole profondità tecnica che possono essere letti come testi a sé, indipendentemente dall'argomentazione principale. Si raccomanda di considerarle lettura opzionale per lettori specialisti.

Una nota sull'Appendice Tecnica IV: le sezioni IV.11–IV.15, che sviluppano la backpropagation come funtore, i transformer come comoduli, e le connessioni con le estensioni di Kan, rappresentano un contributo originale di geometria categoriale applicata all'AI che merita trattazione separata. Nel contesto di questa monografia, si consiglia di leggerle come addendum specialistico indipendente.

Parte Prima — La crisi fondazionale: cosa hanno davvero dimostrato Gödel e Turing

I.1 La posta in gioco nel programma hilbertiano

Per capire cosa hanno demolito i teoremi di Gödel del 1931 e il risultato di Turing del 1936, bisogna capire con precisione cosa intendevano costruire. Il programma di Hilbert — esposto nella sua forma più articolata nei lavori degli anni Venti, con la collaborazione di Ackermann e Bernays — non era semplicemente la ricerca di un fondamento per la matematica. Era qualcosa di più ambizioso e di più specifico: la dimostrazione che la matematica è un sistema chiuso, completo e decidibile.

Questi tre aggettivi hanno contenuto tecnico preciso. *Chiuso* significa coerente: non si può derivare dal sistema sia una proposizione che la sua negazione. *Completo* significa che ogni proposizione matematica ben formata è o dimostrabile o refutabile all'interno del sistema: non esistono verità irraggiungibili con i mezzi formali disponibili. *Decidibile* significa che esiste un procedimento meccanico — un algoritmo, nel linguaggio che Turing avrebbe formalizzato qualche anno dopo — in grado di determinare, per ogni proposizione ben formata, se essa è dimostrabile o no.

Il programma hilbertiano aveva una motivazione filosofica profonda che va al di là della tecnica: liberare la matematica dall'incertezza epistemica prodotta dai paradossi. Il paradosso di Russell, scoperto nel 1901, aveva mostrato che la teoria ingenua degli insiemi di Cantor era contraddittoria: l'insieme di tutti gli insiemi che non contengono se stessi come elementi — l'insieme R tale che $x \in R \leftrightarrow x \notin x$ — genera la contraddizione immediata $R \in R \leftrightarrow R \notin R$. La risposta di Russell e Whitehead nei tre volumi dei *Principia Mathematica* (1910–1913) era una teoria tipizzata degli insiemi che evitava il paradosso per stipulazione: gli insiemi erano organizzati in una gerarchia di tipi in cui un insieme di tipo n non può

contenere elementi di tipo n o superiore. La soluzione era tecnicamente efficace ma filosoficamente insoddisfacente: escludeva il paradosso per decreto, non per comprensione della sua origine.

Hilbert voleva qualcosa di più: voleva *dimostrare*, con metodi finitari e intuitivamente sicuri, che il sistema assiomatico della matematica — qualunque sistema adeguato a contenere l'aritmetica elementare — è coerente. La dimostrazione doveva impiegare soltanto strumenti matematici elementari, primitivi, non soggetti a dubbio: la combinazione finita di simboli, il conteggio, la sostituzione. Hilbert chiamava questo programma *Beweistheorie*, teoria della dimostrazione, e il metodo *Metamathematik*: la matematica applicata alla matematica stessa, studiando le dimostrazioni come oggetti finiti manipolabili.

Tra il 1929 e il 1931, nel giro di due anni, il programma collassò su se stesso. Nel 1929, Kurt Gödel aveva dimostrato il teorema di completezza per la logica del primo ordine: ogni formula valida in tutti i modelli è dimostrabile nel calcolo dei predicati. Era un successo. Ma nel 1930, annunciato a una conferenza a Königsberg e pubblicato nel gennaio del 1931 con il titolo *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, Gödel pubblicò i due teoremi di incompletezza che demolirono il programma hilbertiano in modo definitivo.

I.2 Il primo teorema di incompletezza: la costruzione

La dimostrazione di Gödel è una delle costruzioni intellettuali più ingegnose della storia del pensiero, e merita di essere seguita nei dettagli tecnici anziché parafrasata, perché la parafrase perde invariabilmente il punto essenziale.

Sia S un sistema formale coerente che soddisfa le seguenti condizioni: S è capace di rappresentare l'aritmetica elementare — addizione,

moltiplicazione, e la relazione d'ordine sui naturali — e il suo insieme di assiomi è ricorsivamente enumerabile, ossia esiste un algoritmo che genera tutti e soli gli assiomi di S in ordine crescente. Queste condizioni sono soddisfatte da sistemi come l'aritmetica di Peano (PA) e dalla teoria degli insiemi di Zermelo-Fraenkel (ZF).

Il primo passo fondamentale è la *Gödelianizzazione*, o aritmetizzazione della sintassi. Gödel osservò che ogni simbolo del vocabolario di S può essere associato a un numero naturale, e ogni formula — stringa finita di simboli — può essere associata a un numero naturale attraverso una codifica che oggi chiamiamo *numero di Gödel*. Sia una codifica concreta: assegniamo ai simboli primitivi i numeri primi 2, 3, 5, 7, 11, 13, ... in qualche ordine fisso. Una formula è una sequenza finita di simboli $s_1 s_2 \dots s_n$, a cui viene assegnato il numero di Gödel:

$$\ulcorner s_1 s_2 \dots s_n \urcorner = 2^{g(s_1)} \cdot 3^{g(s_2)} \cdot 5^{g(s_3)} \cdot \dots \cdot p_n^{g(s_n)}$$

dove $g(s_i)$ è il numero assegnato al simbolo s_i e p_n è l' n -esimo numero primo. Per il teorema fondamentale dell'aritmetica, questa codifica è iniettiva: numeri di Gödel diversi corrispondono a formule diverse. Analogamente, una dimostrazione — sequenza finita di formule — riceve un numero di Gödel come prodotto di potenze di primi con esponenti uguali ai numeri di Gödel delle formule componenti.

La conseguenza cruciale è che ogni relazione sintattica tra formule e dimostrazioni — la relazione φ è un assioma, la relazione ψ si ottiene da φ_1 e φ_2 per *modus ponens*, la relazione d è una dimostrazione di φ — diventa una relazione aritmetica tra i numeri di Gödel corrispondenti. Gödel dimostrò che queste relazioni aritmetiche sono tutte *primitive ricorsive*, cioè calcolabili attraverso funzioni definite per composizione e ricorsione primitiva. La relazione più importante è:

$\text{Bew}(d, \varphi) \leftrightarrow$ "il numero d è il numero di Gödel di una dimostrazione della formula con numero di Gödel φ "

Questa relazione è primitiva ricorsiva, dunque rappresentabile all'interno di S : esiste una formula $\text{Bew}(d, y)$ di S con due variabili libere tale che $\text{Bew}(d, \varphi)$ è vera nell'aritmetica standard se e solo se $S \vdash \text{Bew}(n, m)$, dove n e m sono i numerali che rappresentano d e φ .

Sia $\text{Prov}(y)$ la formula $\exists d \text{ Bew}(d, y)$: "esiste una dimostrazione della formula con numero di Gödel y ". Questa è la formula di *dimostrabilità* di S : $\text{Prov}(\ulcorner \varphi \urcorner)$ afferma, all'interno del linguaggio di S , che la formula φ è dimostrabile in S .

Il secondo passo è il *lemma diagonale*, noto anche come lemma di punto fisso. Esso afferma che per ogni formula $F(y)$ di S con una variabile libera esiste una formula G tale che:

$$S \vdash G \leftrightarrow F(\ulcorner G \urcorner)$$

La dimostrazione del lemma è costruttiva. Sia $\text{sub}(x, y)$ la funzione primitiva ricorsiva che, dato il numero di Gödel x di una formula $\varphi(v)$ e un numero n , calcola il numero di Gödel della formula $\varphi(\bar{n})$ ottenuta sostituendo la variabile v con il numerale $\bar{n}$. Sia $D(y)$ la formula $F(\text{sub}(y, y))$. Il numero di Gödel di D sia d . Sia G la formula $D(\bar{d})$, ossia $F(\text{sub}(\bar{d}, \bar{d}))$. Ora: $\text{sub}(d, d)$ è il numero di Gödel della formula $D(\bar{d})$, che è G stesso. Quindi:

$$G = F(\text{sub}(\bar{d}, \bar{d})) \text{ e } \text{sub}(d, d) = \ulcorner G \urcorner$$

da cui $S \vdash G \leftrightarrow F(\ulcorner G \urcorner)$. Il lemma è dimostrato.

Applicando il lemma diagonale alla formula $\neg \text{Prov}(y)$, si ottiene la formula gödeliana: una formula G tale che:

$$S \vdash G \leftrightarrow \neg \text{Prov}(\ulcorner G \urcorner)$$

G afferma di se stessa, nel linguaggio di S, di non essere dimostrabile in S. Si tratta di una variazione aritmetica del paradosso del mentitore — "questa proposizione è falsa" — ma con una differenza cruciale: la formula G è perfettamente ben formata e priva di contraddizione, perché afferma la *non-dimostrabilità*, non la falsità.

Il teorema di Gödel afferma ora: se S è coerente, allora $S \not\vdash G$ e $S \not\vdash \neg G$.

La dimostrazione della prima parte: supponiamo $S \vdash G$. Allora esiste una dimostrazione di G, quindi $\text{Prov}(\ulcorner G \urcorner)$ è vera nell'aritmetica standard e, per la proprietà di rappresentazione, $S \vdash \text{Prov}(\ulcorner G \urcorner)$. Ma $S \vdash G \leftrightarrow \neg \text{Prov}(\ulcorner G \urcorner)$, quindi $S \vdash \neg \text{Prov}(\ulcorner G \urcorner)$. Il sistema S dimostra sia $\text{Prov}(\ulcorner G \urcorner)$ che $\neg \text{Prov}(\ulcorner G \urcorner)$, contraddizione con la coerenza. Dunque $S \not\vdash G$.

La dimostrazione della seconda parte: poiché $S \not\vdash G$, la formula $\text{Prov}(\ulcorner G \urcorner)$ è falsa nell'aritmetica standard. Se S fosse ω -coerente — una condizione più forte della coerenza ordinaria, che afferma che il sistema non può dimostrare $\exists x \neg \varphi(x)$ se dimostra $\varphi(0), \varphi(1), \varphi(2), \dots$ per ogni φ — allora $S \not\vdash \neg G$. In termini più elementari: la formula G è vera nell'aritmetica standard — perché G afferma di non essere dimostrabile in S, e abbiamo mostrato che G è effettivamente non dimostrabile — ma non è dimostrabile in S. S è incompleto.

I.3 Il secondo teorema e la morte del programma hilbertiano

Il secondo teorema di incompletezza è la conseguenza più devastante per il programma di Hilbert. Sia $\text{Con}(S)$ la formula aritmetica che esprime la coerenza di S: formalmente, $\text{Con}(S)$ è la formula $\neg \text{Prov}(\ulcorner 0=1 \urcorner)$, "la formula $0=1$ non è dimostrabile in S". Allora:

Se S è coerente, allora $S \not\vdash \text{Con}(S)$.

La dimostrazione sfrutta la struttura della dimostrazione del primo teorema. Abbiamo mostrato che la sequenza ragionativa "se S è coerente allora $S \not\vdash G$ " è formalizzabile in S : cioè $S \vdash \text{Con}(S) \rightarrow \neg \text{Prov}(\ulcorner G \urcorner)$. Ma per il lemma diagonale $S \vdash G \leftrightarrow \neg \text{Prov}(\ulcorner G \urcorner)$, quindi $S \vdash \text{Con}(S) \rightarrow G$. Se $S \vdash \text{Con}(S)$, allora $S \vdash G$ — ma abbiamo già mostrato che quest'ultima conseguenza viola la coerenza di S . Dunque $S \not\vdash \text{Con}(S)$.

Questo significa: S non può dimostrare la propria coerenza, a meno di essere incoerente. Il sogno hilbertiano di fondare la matematica su una base finitariamente sicura e dimostrabilmente coerente è impossibile. Per dimostrare la coerenza di S occorre un sistema S' più forte di S — ma allora occorre dimostrare la coerenza di S' , e così via. La fondazione si allontana indefinitamente.

I.4 Il problema della fermata e la frontiera della computazione

Cinque anni dopo i teoremi di Gödel, Alan Turing dimostrò un risultato indipendente ma strutturalmente analogo, che completò la distruzione del programma hilbertiano da una direzione diversa. Turing stava affrontando il *Entscheidungsproblem* di Hilbert e Ackermann: esiste un procedimento meccanico che, data una qualunque formula della logica del primo ordine, determini in un numero finito di passi se essa è universalmente valida?

Per rispondere, Turing costruì un modello formale di *procedimento meccanico*: la macchina di Turing. Una macchina di Turing è una quintupla $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ dove Q è un insieme finito di stati, Σ è un alfabeto finito di simboli (incluso il simbolo vuoto $\square$), $\delta: Q \times \Sigma \rightarrow Q \times \Sigma \times \{L, R\}$ è la funzione di transizione parziale, $q_0 \in Q$ è lo stato iniziale, e $F \subseteq Q$ è l'insieme degli stati accettanti. La macchina opera su un nastro

infinito in entrambe le direzioni, leggendo e scrivendo simboli e spostandosi a sinistra o a destra secondo δ .

Un calcolo della macchina M su input x è la sequenza delle configurazioni successive — coppie $\langle q, w \rangle$ di stato corrente e contenuto del nastro — a partire dalla configurazione iniziale $\langle q_0, x \rangle$. Il calcolo può terminare in uno stato accettante (M accetta x), in uno stato non accettante (M rifiuta x), oppure non terminare mai (M diverge su x).

Turing formulò il *problema della fermata*: esiste una macchina di Turing H che, data la descrizione $\lceil M \rceil$ di una macchina di Turing M e un input x , risponda "sì" se M termina su x e "no" se M non termina su x ?

La dimostrazione che H non può esistere è per diagonalizzazione. Supponiamo che H esista e costruiamo la macchina D che, data la descrizione $\lceil M \rceil$ di una macchina, si comporta così:

D su input $\lceil M \rceil$: simula H su $\langle \lceil M \rceil, \lceil M \rceil \rangle$; se H risponde "sì" (M termina su $\lceil M \rceil$), allora D diverge; se H risponde "no" (M non termina su $\lceil M \rceil$), allora D termina.

Chiediamoci ora cosa fa D sul proprio input $\lceil D \rceil$:

Se D termina su $\lceil D \rceil$, allora H risponde "sì" su $\langle \lceil D \rceil, \lceil D \rceil \rangle$, quindi D diverge su $\lceil D \rceil$. Contraddizione. Se D diverge su $\lceil D \rceil$, allora H risponde "no" su $\langle \lceil D \rceil, \lceil D \rceil \rangle$, quindi D termina su $\lceil D \rceil$. Contraddizione.

In entrambi i casi si raggiunge una contraddizione: H non può esistere.

I.5 La struttura condivisa delle due crisi

I risultati di Gödel e Turing condividono una struttura matematica profonda che la vulgata non articola con sufficiente precisione. Entrambi impiegano la *diagonalizzazione*: costruiscono un oggetto che si

autorimanda — la formula G che parla di se stessa, la macchina D che simula se stessa — e mostrano che questo autoreferimento genera una contraddizione se si assume l'esistenza di un oggetto onnisciente (il sistema completo, la macchina H). La diagonalizzazione di Cantor per la non numerabilità dei reali, la costruzione di Russell per il paradosso degli insiemi, il lemma diagonale di Gödel e la costruzione di Turing sono tutte istanze della medesima tecnica: costruire un oggetto che è diverso da ogni elemento di una lista sistematica, negando la possibilità di una lista completa.

Questa struttura comune rivela qualcosa di più profondo: i sistemi formali hanno frontiere strutturali che non possono essere attraversate dall'interno. Un sistema formale non può dimostrare la propria coerenza; un algoritmo non può decidere la propria terminazione. La frontiera non è un limite tecnico da superare con sistemi più potenti: è una proprietà topologica dello spazio degli enunciati e dei calcoli, una proprietà che si riproduce a qualunque livello di potenza del sistema considerato.

Ma questa topologia ha una conseguenza che il programma hilbertiano non aveva previsto e che è al centro dell'argomento di questa monografia: i limiti dei sistemi formali non sono limiti dell'intelligenza. Gödel stesso, nella sua formula G , comprendeva che G era vera pur essendo indimostrabile nel sistema. Turing, ragionando sulla macchina D , comprendeva la struttura della contraddizione che la macchina produceva. L'intelligenza che costruisce i sistemi formali è capace di vedere al di là dei limiti di quei sistemi — non perché sia infinita o soprannaturale, ma perché opera con meccanismi diversi dalla derivazione formale.

Questo punto è stato sviluppato, in forme variamente eccentriche, da Roger Penrose nei volumi *The Emperor's New Mind* (1989) e *Shadows of the*

Mind (1994), con la tesi che i meccanismi neurali sfruttino effetti quantistici non computabili. La tesi penrosiana ha ricevuto critiche severe e in larga misura giustificate, ma il suo punto di partenza — che la comprensione umana sembri eccedere la derivazione formale — era genuinamente motivato dai teoremi di Gödel. La risposta corretta all'osservazione di Penrose non è negare il fenomeno, ma spiegarlo senza ricorrere a meccanismi non computabili: un compito che riprenderemo nella Parte Quinta, quando avremo costruito gli strumenti necessari.

I.6 La decidibilità parziale e la gerarchia di Church-Turing

I risultati di incompletezza e indecidibilità non dicono che tutto è irraggiungibile: specificano con precisione ciò che è raggiungibile. La gerarchia computazionale che emerge dal lavoro di Kleene, Post e i loro successori — nota come gerarchia aritmetica — classifica gli insiemi di numeri naturali secondo la complessità delle proprietà che li definiscono.

Un insieme A è *ricorsivo* (decidibile) se esiste una macchina di Turing che, su ogni input, termina e risponde correttamente "sì" se il numero è in A e "no" altrimenti. L'insieme dei teoremi di un sistema formale coerente e con assiomi ricorsivamente enumerabili non è ricorsivo — questo è ciò che il problema della fermata implica per i sistemi logici. Ma è *ricorsivamente enumerabile* (RE): esiste una macchina di Turing che, su ogni input, termina e risponde "sì" se il numero è in A , ma può divergere se il numero non è in A . Il processo di ricerca di una dimostrazione è semidecidibile: se una dimostrazione esiste, la si trova in tempo finito; se non esiste, la ricerca non termina.

I livelli superiori della gerarchia aritmetica classificano problemi progressivamente più complessi: Σ_1 (RE), Π_1 (co-RE), Σ_2 , Π_2 , ... fino all'oracolo per il problema della fermata e ai suoi successori transfiniti.

Questa gerarchia è strettamente crescente: a ogni livello esistono problemi che non possono essere risolti con le risorse dei livelli inferiori.

La gerarchia aritmetica stabilisce non solo dove si trova il confine della computabilità, ma *come* quel confine è strutturato: è una stratificazione infinita, non una linea netta. L'incompletezza di Gödel non è la scoperta di un abisso insondabile: è la scoperta che lo spazio dei problemi matematici ha una struttura ordinata che nessun sistema formale singolo può esaurire interamente. Il che porta direttamente alla domanda che questa monografia si propone di rispondere: se il sistema formale è lo strumento sbagliato per l'intelligenza generale, quale strumento usa il cervello?

Parte Seconda — Il logicismo cognitivo: l'ipotesi del sistema fisico di simboli e il suo collasso

II.1 Newell e Simon: la filosofia travestita da ingegneria

Nel 1976, Allen Newell e Herbert Simon ricevettero il premio Turing dell'Association for Computing Machinery. Il discorso con cui Simon accompagnò il riconoscimento — pubblicato l'anno successivo con il titolo *Computer Science as Empirical Inquiry: Symbols and Search* — contiene una delle tesi più ambiziose mai formulate nelle scienze cognitive, presentata con la fiducia di chi aveva già costruito i sistemi che sembravano confermarne la verità. La tesi è la *Physical Symbol System Hypothesis*, che tradurrò come ipotesi del sistema fisico di simboli (HFSS).

Nella formulazione di Newell e Simon:

"Un sistema fisico di simboli possiede i mezzi necessari e sufficienti per l'azione intelligente generale."

Questa formulazione è notevole per la sua compressione. Per comprenderla occorre svilupparne i componenti. Un *sistema fisico di simboli* è un sistema che: (a) contiene entità fisiche discrete — i simboli — che possono essere create, distrutte, copiate, trasformate e combinate; (b) ha strutture simboliche, ossia simboli che fungono da designatori di altri simboli o di strutture di simboli; (c) ha un repertorio di processi — una macchina di Turing universale è l'esempio canonico, ma anche i sistemi di riscrittura, le algebre di termini, la logica del primo ordine — capaci di leggere, scrivere e modificare le strutture simboliche. Il termine *fisico* enfatizza che il sistema deve essere implementato in un substrato fisico reale, non essere semplicemente un'astrazione matematica.

L'ipotesi afferma che tali sistemi, e *soltanto* tali sistemi, sono capaci di azione intelligente generale. Il verso dell'implicazione nella direzione dell'intelligenza — l'intelligenza richiede un sistema fisico di simboli — è l'affermazione più forte e più controversa. Essa implica che qualunque sistema che esibisca intelligenza genuina sia, sotto le apparenze, un sistema di manipolazione simbolica. I cervelli umani sono sistemi fisici di simboli; le formiche, se sono intelligenti, sono sistemi fisici di simboli; qualunque extraterrestre intelligente è un sistema fisico di simboli. L'intelligenza ha un'unica architettura profonda.

La HFSS non era soltanto una tesi filosofica: era anche il programma di ricerca di un'intera generazione dell'informatica. I sistemi esperti degli anni Settanta e Ottanta — MYCIN per la diagnosi delle infezioni batteriche, DENDRAL per l'analisi delle strutture molecolari da spettri di massa, R1/XCON per la configurazione dei computer VAX di Digital Equipment Corporation — erano implementazioni concrete dell'ipotesi. La conoscenza di un dominio veniva rappresentata come un insieme di regole simboliche del tipo SE ... ALLORA ..., e l'inferenza era l'applicazione sistematica di queste regole a una base di fatti simbolici. Il

processo inferenziale era trasparente, ispezionabile, e tracciabile: si poteva seguire esattamente quale catena di regole aveva condotto a una determinata conclusione.

MYCIN, sviluppato da Edward Shortliffe a Stanford tra il 1972 e il 1974, è il caso più studiato. Il sistema conteneva circa 600 regole di produzione della forma:

SE il sito del campione è sangue E l'organismo è gram-positivo E l'organismo è a forma di cocco E la disposizione degli organismi è in catene ALLORA c'è evidenza favorevole (0.7) che l'organismo sia streptococcus

La struttura era elegante e il sistema funzionava: in valutazioni comparative condotte da Victor Yu e collaboratori nel 1979, MYCIN eguagliava o superava le diagnosi di medici specialisti nel trattamento delle meningiti batteriche. L'HFSS sembrava avere una conferma sperimentale diretta.

II.2 Le fondamenta filosofiche dell'HFSS

L'ipotesi di Newell e Simon non era nata nel vuoto: si inseriva in una tradizione filosofica precisa, quella del *computazionalismo classico*, che identificava la cognizione con la computazione simbolica. Le radici di questa posizione risalgono almeno all'impostazione carnapiana della filosofia del linguaggio come analisi logica, ma la sua forma cognitiva è meglio individuata nel lavoro di Fodor, in particolare nel volume *The Language of Thought* del 1975.

Fodor aveva argomentato che il pensiero è letteralmente linguaggio: esiste un *linguaggio del pensiero* — il *mentalese* — le cui unità sono rappresentazioni mentali con struttura compositiva, e i processi cognitivi sono computazioni su queste rappresentazioni. Il pensiero è

manipolazione di simboli con struttura sintattica, e la semantica dei pensieri — il loro contenuto, il loro riferimento — superviene sulla sintassi. La cognizione è esattamente ciò che la HFSS afferma.

Il computazionalismo classico aveva radici ancora più profonde nella concezione del *problem solving* come ricerca nello spazio degli stati. Newell e Simon avevano elaborato questa concezione già nel lavoro sul General Problem Solver (GPS) del 1959: ogni problema può essere rappresentato come uno spazio di stati possibili, con un operatore di transizione che specifica come si passa da uno stato all'altro, e il problema-solving è la ricerca del percorso nello spazio degli stati che conduce dallo stato iniziale a uno stato-goal. La ricerca può essere guidata da euristiche — funzioni di stima della distanza dallo stato goal — che la rendono più efficiente, ma la struttura profonda è sempre quella della navigazione simbolica nello spazio degli stati.

Questa concezione ha una virtù epistemica importante: è completamente trasparente. Il comportamento del sistema è in linea di principio interamente spiegabile nei termini delle regole che lo governano, senza residuo. Non ci sono processi oscuri, meccanismi subpersonali, contributi non computabili. Se il sistema produce un comportamento, c'è sempre una sequenza di passi simbolici che lo spiega. Questa trasparenza era considerata un vantaggio rispetto alle teorie connessioniste emergenti: la rete neurale, si diceva, è una scatola nera; il sistema simbolico è un libro aperto.

II.3 Il frame problem: non un bug tecnico ma una frattura strutturale

La prima frattura nel programma simbolico apparve non come una confutazione diretta dell'HFSS, ma come un problema tecnico che

sembrava risolvibile con raffinamenti successivi: il *frame problem*, formulato da John McCarthy e Patrick Hayes nel saggio *Some Philosophical Problems from the Standpoint of Artificial Intelligence* del 1969. La formulazione originale è così modesta nei toni da sembrare un problema amministrativo anziché una crisi profonda, e questo ha contribuito a ritardarne il riconoscimento come problema fondamentale.

Il problema si pone nel contesto della logica delle situazioni. Una situazione è un'istantanea del mondo a un dato momento: un insieme di fatti veri in quel momento. Un'azione — spostare un blocco, accendere una luce, aprire una porta — trasforma la situazione s in una situazione successiva s' . Come si descrive formalmente s' data s e l'azione?

La risposta ovvia è specificare gli *effetti* dell'azione: l'azione *spostare*(blocco_1, posizione_2) produce il fatto *at*(blocco_1, posizione_2) e rimuove il fatto *at*(blocco_1, posizione_1). Ma questa specifica degli effetti non dice nulla di tutto ciò che *non cambia*: il blocco B è ancora nella posizione 3, la luce è ancora accesa, la porta è ancora chiusa. Per ragionare su s' occorre sapere non solo cosa è cambiato, ma anche tutto ciò che è rimasto invariato. E ciò che rimane invariato è il resto del mondo: potenzialmente ogni fatto non coinvolto nell'azione.

La soluzione immediata è aggiungere *assiomi di frame*: per ogni azione A e per ogni fatto F non nel dominio di effetti di A , si specifica che F è invariato da A . Ma per un dominio con n azioni e m fatti, questo richiede $O(n \cdot m)$ assiomi di frame — uno per ogni coppia azione-fatto. Per un dominio realistico, n e m sono entrambi molto grandi, e il numero di assiomi di frame esplode combinatorialmente.

Questo non sarebbe ancora fatale se non fosse per un problema più profondo: come si stabilisce, per una data azione, quali fatti essa *non* cambia? Per azioni fisiche semplici — spostare un oggetto — il dominio

di non-effetti è chiaramente il complemento del dominio di effetti. Ma per azioni con effetti causali a catena — accendere un motore, aprire una valvola in un impianto interconnesso, pronunciare una parola in un contesto sociale — il dominio di effetti è in linea di principio illimitato e non specificabile in anticipo. Non esiste un criterio generale che, data un'azione, determini a priori tutti e soli i fatti che essa non cambia.

McCarthy propose la *Circumscription* come soluzione: un principio di ragionamento non monotono che, data un'azione con effetti specificati, conclude per *minimizzazione* che tutti gli altri fatti rimangono invariati. Formalmente, la circumscription di un predicato P in una formula φ è la formula che afferma che l'estensione di P è *minimale* tra tutte le estensioni che rendono φ vera. Applicata al frame problem, la circumscription dice: degli effetti dell'azione A , *assume per default* che siano solo quelli esplicitamente specificati.

La soluzione è tecnicamente elegante ma non risolve il problema fondamentale: la circumscription è un operatore di secondo ordine (quantifica su predicati), e il ragionamento per circumscription è computazionalmente intrattabile nel caso generale — Π_2 -completo nella gerarchia polinomiale, come dimostrò Vladimir Lifschitz nel 1985. Per i problemi pratici di dimensione realistica, il ragionamento per circumscription è troppo costoso per essere impiegato.

Reiter (1980) propose la *closed world assumption* (CWA): ciò che non è noto essere vero è assunto essere falso. La CWA è computazionalmente più trattabile, ma ha il difetto opposto: assume per default che tutto ciò che non è esplicitamente specificato sia falso, il che porta a conclusioni sbagliate in domini aperti dove la conoscenza è necessariamente incompleta.

Il frame problem non è mai stato risolto nell'AI simbolica. Non per mancanza di intelligenza o di tentativi, ma perché il problema tocca qualcosa di strutturale: la rappresentazione esplicita e completa di un dominio aperto richiede una conoscenza esaustiva del dominio, e nessun sistema che apprende per specifica esplicita di regole può acquisire questa conoscenza per via formale. Il mondo non ha una struttura che si lascia descrivere esaustivamente da un insieme finito di regole.

II.4 L'inverno dell'AI e il qualificatore problem

Il frame problem era il più profondo tra i problemi dell'AI simbolica, ma non il solo. Un secondo problema strutturale — il *qualification problem*, formulato anch'esso da McCarthy — riguardava le precondizioni delle azioni. Per specificare un'azione come *avviare un'automobile* occorre elencare le condizioni necessarie: la chiave è inserita, il serbatoio non è vuoto, la batteria è carica, il motore non è guasto, non è -40°C , l'automobile non è sott'acqua, e così via. Queste precondizioni sono potenzialmente infinite: per qualunque elenco finito di condizioni, è concepibile una circostanza anomala in cui l'azione fallisce pur soddisfacendole tutte.

Il problema della qualificazione è la versione diacronica del frame problem: come il frame problem riguarda la stabilità del mondo attraverso le azioni, il qualification problem riguarda l'incertezza sulle precondizioni. Entrambi hanno la stessa struttura: la rappresentazione simbolica esplicita è sempre incompleta rispetto alla complessità del mondo.

Un terzo problema era il *ramification problem*: quando un'azione ha effetti indiretti attraverso catene causali, come si calcolano tutti gli stati risultanti? Se sposto un tavolo, si sposta anche il bicchiere che vi è sopra,

e si sposta anche l'acqua nel bicchiere, e cambia la pressione sull'architrave dove il tavolo era appoggiato, e così via. Il dominio di ramificazione degli effetti di un'azione è in linea di principio illimitato.

Questi tre problemi — *frame*, *qualification*, *ramification* — condividono la struttura di fondo: il mondo è troppo ricco, troppo interconnesso, troppo irregolare per essere rappresentato esaustivamente da simboli e regole. La conoscenza del senso comune — *common sense knowledge* — che gli esseri umani impiegano senza sforzo nelle interazioni quotidiane ha una struttura che non si lascia catturare dalla rappresentazione proposizionale esplicita.

La crisi pratica dell'AI simbolica esplose negli anni Ottanta. I sistemi esperti, inizialmente celebrati, rivelarono la loro fragilità fuori dai domini chiusi per cui erano stati costruiti. MYCIN non poteva trasferire la propria expertise medica a un dominio diverso: la conoscenza era incapsulata in regole specifiche per la diagnosi batterica e non generalizzava. La manutenzione delle basi di conoscenza diventava progressivamente più costosa man mano che il dominio cresceva: ogni nuova regola poteva interagire con le regole esistenti in modi imprevedibili, producendo incoerenze che richiedevano correzioni manuali.

Il Consiglio Britannico per la Ricerca in Informatica (SERC) e la Defense Advanced Research Projects Agency (DARPA) negli Stati Uniti ridussero drasticamente i finanziamenti all'AI simbolica tra il 1987 e il 1991 — quello che i commentatori chiamarono il "secondo inverno dell'AI". Non per mancanza di risultati, ma per mancanza di generalizzabilità: i sistemi simbolici funzionavano in laboratorio, non nel mondo.

II.5 Il nodo filosofico: la rappresentazione come specchio o come strumento

La crisi dell'AI simbolica ha una lettura filosofica che va al di là della storia dell'informatica. L'HFSS incorporava implicitamente una teoria della rappresentazione: la conoscenza è rappresentazione esplicita di stati del mondo attraverso simboli con contenuto semantico determinato. Questa teoria è la versione computazionale della *picture theory of meaning* del Wittgenstein del *Tractatus Logico-Philosophicus* (1921): una proposizione è un fatto che funge da immagine di un altro fatto, e la struttura logica della proposizione deve rispecchiare la struttura logica del fatto che rappresenta.

Il Wittgenstein del *Tractatus* aveva già articolato, con grande lucidità, il limite di questa visione: la proposizione può rappresentare qualsiasi struttura di fatti, ma non può rappresentare se stessa come rappresentazione — non può dire ciò che mostra. Il confine del linguaggio — il confine del rappresentabile — è inesprimibile dall'interno. Questa intuizione è formalmente imparentata con i teoremi di Gödel: la struttura dell'autoreferimento che genera l'incompletezza è la stessa che genera l'indicibilità dei fondamenti nel *Tractatus*.

Il Wittgenstein delle *Philosophische Untersuchungen* (1953) aveva poi demolito la picture theory da una direzione diversa: il significato non è determinato dalla struttura logica ma dall'uso nelle pratiche sociali. Non esiste un'essenza del significato indipendente dal contesto d'uso: i giochi linguistici sono troppo vari, troppo aperti, troppo sensibili al contesto per essere catturati da una struttura logica uniforme.

Il frame problem è la versione operativa della critica wittgensteiniana alla picture theory: la rappresentazione simbolica esplicita non può catturare la conoscenza del senso comune perché questa conoscenza è

costitutivamente contestuale, aperta, sensibile a variazioni che nessuna lista finita di regole può anticipare. Il problema non è che non abbiamo abbastanza regole: è che la struttura della conoscenza pratica è radicalmente diversa dalla struttura della rappresentazione proposizionale.

Vale la pena segnalare, anche solo per cenni, che questa diagnosi non è esclusivamente una critica alla tradizione dell'AI: è una critica alla stessa impostazione del Circolo di Vienna, e in particolare al programma neopositivista di riduzione di tutta la conoscenza significativa a proposizioni verificabili. Carnap aveva esplicitamente tentato di costruire una logica induttiva che potesse catturare la struttura della conoscenza empirica nello stesso modo in cui la logica deduttiva cattura la struttura delle dimostrazioni matematiche. Il fallimento di questo programma — documentato da Quine in *Two Dogmas of Empiricism* (1951) e sviluppato in direzione cognitiva da Dreyfus — ha la stessa struttura del frame problem: la conoscenza pratica del mondo non ammette formalizzazione esaustiva. Questa connessione tra la crisi del neopositivismo e la crisi dell'AI simbolica verrà sviluppata in modo più sistematico nell'Espansione B.7, dove il fallimento del programma carnappiano di logica induttiva viene messo in parallelo con il qualificatore problem dei sistemi esperti.

Questa diagnosi fu sviluppata in modo influente da Hubert Dreyfus, filosofo fenomenologico che dedicò decenni di lavoro alla critica dell'AI simbolica, a partire dal volume *What Computers Can't Do* del 1972. Dreyfus argomentava, sulla base di Heidegger e Merleau-Ponty, che l'intelligenza pratica umana non è fondata sulla rappresentazione esplicita di regole ma sull'immersione in un mondo di pratiche incorporate: il *Dasein* heideggeriano non ragiona su regole per navigare il mondo, vi abita. La critica di Dreyfus fu respinta con qualche disprezzo dalla

comunità dell'AI simbolica, ma i fatti le diedero ragione in modo progressivo e inesorabile.

II.6 Cosa sopravvive del programma simbolico

Sarebbe sbagliato concludere che il collasso dell'AI simbolica sia la confutazione di tutto il programma logicista. Il ragionamento simbolico è reale e potente in domini chiusi e ben definiti: la verifica formale di software e hardware, la pianificazione automatica in ambienti controllati, la dimostrazione automatica di teoremi in matematica, i solutori SAT e SMT impiegati nell'industria. Tutti questi sono domini in cui il mondo è abbastanza semplice, strutturato e stabile da consentire la rappresentazione simbolica completa.

Il punto non è che la logica formale sia inutile: è che ha un dominio di applicabilità preciso, che non include la cognizione aperta nel mondo reale. La logica è uno strumento potente per i problemi che si lasciano rappresentare completamente in termini simbolici — e molti problemi importanti hanno questa proprietà. La tesi di questa monografia è più circoscritta: la logica non è il meccanismo fondamentale dell'intelligenza, nemmeno dell'intelligenza umana. È uno strumento che l'intelligenza ha sviluppato per un sottoinsieme ristretto e particolarmente ben strutturato dei suoi problemi.

Questa distinzione — tra lo strumento e il meccanismo — è il cuore di ciò che si intende sviluppare nelle parti successive, dove si esamina quale meccanismo abbia effettivamente prodotto l'intelligenza biologica e quale matematica governi i sistemi artificiali che l'hanno approssimata con successo negli ultimi vent'anni.

Parte Terza — La matematica dell'intelligenza artificiale attuale: ottimizzazione, gradiente e attenzione

III.1 Il cambio di paradigma come cambio di problema

Prima di entrare nelle derivazioni, vale la pena precisare la natura del mutamento che ha portato dall'AI simbolica all'AI statistica e neurale. Non si tratta semplicemente di un cambio di tecnica: è un cambio nella formulazione del problema fondamentale.

L'AI simbolica si poneva il problema della *rappresentazione*: come codificare la conoscenza del dominio in un sistema di regole formali che un motore inferenziale potesse applicare? Il problema era epistemologico: si trattava di estrarre la conoscenza dagli esperti umani e trascriverla in simboli. L'intelligenza veniva concepita come un repertorio di conoscenze esplicite più un meccanismo di applicazione.

L'AI statistica si pone il problema dell'*ottimizzazione*: data una famiglia parametrizzata di funzioni e un insieme di esempi di comportamento desiderato, trovare i parametri che minimizzano la discrepanza tra l'output della funzione e l'output desiderato. Il problema è matematico: si tratta di navigare uno spazio di alta dimensione verso un minimo di una funzione di perdita. La conoscenza non è rappresentata esplicitamente: è distribuita nei parametri del sistema, e il sistema non può, in generale, essere interrogato su ciò che sa.

Questa differenza di impostazione ha una conseguenza che è al centro dell'argomentazione di questa monografia: i sistemi neurali ottimizzati per gradient descent *approssimano* le funzioni bersaglio, non le *calcolano esattamente*. L'approssimazione è ottimale nel senso statistico — minimizza la perdita attesa sul dominio di addestramento — ma non è garantita essere corretta su ogni singolo esempio. Questo è radicalmente

diverso dall'architettura simbolica, dove la correttezza di ogni singola derivazione è garantita dalla struttura delle regole.

III.2 La backpropagation: derivazione completa

Il meccanismo fondamentale dell'apprendimento nei sistemi neurali profondi è la *backpropagation*, formalizzata nella sua forma moderna da Rumelhart, Hinton e Williams nel 1986, benché la struttura matematica fosse nota in forma embrionale già nei lavori di Bryson e Ho negli anni Sessanta nel contesto del controllo ottimale. Il punto di partenza è un insieme di addestramento $\mathcal{D} = \{(x_i^{(i)}, y_i^{(i)})\}_{i=1}^N$ di coppie input-output, e una rete neurale $f_{\theta}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ parametrizzata da un vettore di parametri $\theta \in \mathbb{R}^p$.

La rete è composta da L strati, con lo strato l che produce un'attivazione:

$$h^{(l)} = \sigma^{(l)}(W^{(l)} h^{(l-1)} + b^{(l)})$$

dove $W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ è la matrice dei pesi dello strato l , $b^{(l)} \in \mathbb{R}^{d_l}$ è il vettore dei bias, $\sigma^{(l)}$ è una funzione di attivazione non lineare applicata elemento per elemento, $h^{(0)} = x$ è l'input e $h^{(L)} = \hat{y}$ è l'output. L'insieme dei parametri è $\theta = \{W^{(l)}, b^{(l)}\}_{l=1}^L$.

La funzione di perdita $\mathcal{L}(\theta)$ misura la discrepanza tra le previsioni della rete e i valori target sull'intero insieme di addestramento:

$$\mathcal{L}(\theta) = (1/N) \sum_{i=1}^N \ell(f_{\theta}(x^{(i)}), y^{(i)})$$

dove ℓ è una funzione di perdita elementare — tipicamente la perdita quadratica $\ell(\hat{y}, y) = \|\hat{y} - y\|^2$ per la regressione o la perdita di cross-entropia $\ell(\hat{y}, y) = -\sum_k y_k \log \hat{y}_k$ per la classificazione. L'obiettivo è trovare $\theta^* = \operatorname{argmin}_{\theta} \mathcal{L}(\theta)$.

Poiché $\mathcal{L}$ è in generale non convessa e P può essere dell'ordine di 10^9 – 10^{12} nei modelli moderni, la minimizzazione globale è computazionalmente intrattabile. Si ricorre alla discesa del gradiente stocastica (SGD): a ogni passo si estrae un minibatch $\mathcal{P} \subset \mathcal{D}$ di dimensione $B \ll N$ e si aggiornano i parametri nella direzione opposta al gradiente stimato:

$$\theta \leftarrow \theta - \eta \cdot (1/B) \sum_{(x,y) \in \mathcal{P}} \nabla_{\theta} \ell(f_{\theta}(x), y)$$

dove $\eta > 0$ è il tasso di apprendimento. Il calcolo efficiente di $\nabla_{\theta} \ell$ è il compito della backpropagation.

Sia $\ell = \ell(h^{\{l\}}, y)$ la perdita su un singolo esempio. Il gradiente rispetto ai parametri dello strato l si calcola attraverso la regola della catena. Definiamo il *segnale di errore* dello strato l come:

$$\delta^{\{l\}} = \partial \ell / \partial z^{\{l\}}$$

dove $z^{\{l\}} = W^{\{l\}} h^{\{l-1\}} + b^{\{l\}}$ è l'attivazione pre-non linearità dello strato l . La regola della catena dà la relazione ricorrente:

$$\delta^{\{l\}} = \partial \ell / \partial z^{\{l\}} = (\partial z^{\{l+1\}} / \partial h^{\{l\}})^T \cdot (\partial h^{\{l+1\}} / \partial z^{\{l+1\}}) \cdot \delta^{\{l+1\}}$$

Sviluppando i termini: $\partial z^{\{l+1\}} / \partial h^{\{l\}} = W^{\{l+1\}}$, e $\partial h^{\{l+1\}} / \partial z^{\{l+1\}} = \text{diag}(\sigma'^{\{l+1\}}(z^{\{l+1\}}))$ dove σ' è la derivata della funzione di attivazione. Quindi:

$$\delta^{\{l\}} = (W^{\{l+1\}})^T \cdot \text{diag}(\sigma'^{\{l+1\}}(z^{\{l+1\}})) \cdot \delta^{\{l+1\}}$$

Questa è la formula di *retropropagazione* degli errori: il segnale di errore allo strato l dipende dal segnale di errore allo strato $l+1$ pre-moltiplicato per la trasposta della matrice dei pesi e pesato dalla derivata della nonlinearity. Il flusso dell'informazione è duale rispetto al flusso del

calcolo in avanti: mentre la forward pass propaga l'input verso l'output, la backward pass propaga il gradiente dall'output verso l'input.

I gradienti rispetto ai parametri sono infine:

$$\partial \ell / \partial \mathbf{W}^{\{l\}} = \delta^{\{l\}} \cdot (\mathbf{h}^{\{l-1\}})^T$$

$$\partial \ell / \partial \mathbf{b}^{\{l\}} = \delta^{\{l\}}$$

La complessità computazionale dell'intera backward pass è della stessa classe della forward pass — $O(\sum_l d_l \cdot d_{l-1})$ — il che rende la backpropagation scalabile a reti di dimensione arbitraria.

III.3 Il problema del gradiente che svanisce e le sue soluzioni

La backpropagation ha un problema strutturale che ha ostacolato l'addestramento di reti profonde per quasi due decenni: il *vanishing gradient problem*. Per capirne la natura, si osservi che la formula ricorrente per $\delta^{\{l\}}$ moltiplica iterativamente la derivata della nonlinearietà σ' e i pesi $\mathbf{W}^{\{l+1\}}$ per ogni strato. Se il prodotto spettrale $\rho(\mathbf{W}^{\{l+1\}} \cdot \text{diag}(\sigma'^{\{l+1\}})) < 1$ — ossia se il raggio spettrale del prodotto delle matrici Jacobiane è inferiore a uno — allora il segnale di errore si attenua esponenzialmente al diminuire di l . Per reti con $L > 10$ strati, i gradienti agli strati profondi diventano numericamente indistinguibili da zero, e l'addestramento si blocca.

Con funzioni di attivazione sigmoidi $\sigma(z) = 1/(1+e^{-z})$ — standard fino ai primi anni 2000 — il problema è acuto perché $\sigma'(z) = \sigma(z)(1-\sigma(z)) \leq 1/4$ per ogni z , e il prodotto iterato di fattori $\leq 1/4$ svanisce esponenzialmente. La soluzione principale è l'uso della funzione ReLU (Rectified Linear Unit), introdotta da Nair e Hinton (2010) e teorizzata da Glorot, Bordes e Bengio (2011):

$$\sigma(z) = \max(0, z)$$

la cui derivata è $\sigma'(z) = \mathbb{1}[z > 0]$ — uno o zero, mai un fattore di attenuazione frazionario per $z > 0$. La ReLU non elimina il vanishing gradient — i neuroni con $z \leq 0$ trasmettono gradiente nullo, il problema dei *neuroni morti* — ma lo attenua drasticamente nella pratica, consentendo l'addestramento di reti con centinaia di strati.

Una soluzione complementare sono le *connessioni residue*, introdotte da He, Zhang, Ren e Sun nel 2016 con le ResNet: ogni blocco di strati calcola non $h^{\{l+1\}} = f(h^{\{l\}})$ ma $h^{\{l+1\}} = h^{\{l\}} + f(h^{\{l\}})$, aggiungendo un percorso diretto che preserva il gradiente attraverso la somma. Formalmente, il gradiente attraverso una connessione residuale è:

$$\partial h^{\{l+1\}} / \partial h^{\{l\}} = I + \partial f / \partial h^{\{l\}}$$

dove I è la matrice identità: il termine identità garantisce che il gradiente non possa svanire completamente anche se $\partial f / \partial h^{\{l\}}$ è piccolo. Le ResNet hanno permesso l'addestramento di reti con più di 1000 strati, e la struttura residuale è diventata il componente architetturale dominante nei modelli profondi.

III.4 Il meccanismo di attenzione: derivazione e struttura

Il meccanismo di attenzione, introdotto da Bahdanau, Cho e Bengio nel 2015 per la traduzione automatica e portato alla sua forma canonica da Vaswani e colleghi nel fondamentale articolo *Attention Is All You Need* del 2017, è il componente tecnico che più di ogni altro ha determinato il salto qualitativo delle prestazioni nei modelli linguistici degli ultimi anni. Vale la pena derivarne la struttura con cura perché essa rivela qualcosa di non banale sulla natura dell'elaborazione dell'informazione che questi sistemi implementano.

Il punto di partenza è una sequenza di vettori di input $X = [x_1, x_2, \dots, x_n]$ $\in \mathbb{R}^{n \times d}$, dove n è la lunghezza della sequenza e d è la dimensione dei vettori. Ogni posizione della sequenza deve produrre un vettore di output che integri informazioni dalle altre posizioni in modo pesato dalla rilevanza contestuale.

L'attenzione *scaled dot-product* opera come segue. Dati tre insiemi di proiezioni lineari — query $Q = X W_Q$, key $K = X W_K$, value $V = X W_V$ — dove $W_Q, W_K \in \mathbb{R}^{d \times d_k}$ e $W_V \in \mathbb{R}^{d \times d_v}$ sono matrici di proiezione apprendibili, l'output dell'attenzione è:

$$\text{Attn}(Q, K, V) = \text{softmax}(Q K^T / \sqrt{d_k}) \cdot V$$

Sviluppiamo il meccanismo elemento per elemento. Per la i -esima posizione della sequenza, la query è $q_i = x_i W_Q \in \mathbb{R}^{d_k}$. Il *punteggio di attenzione* della posizione i rispetto alla posizione j è:

$$s_{ij} = q_i \cdot k_j / \sqrt{d_k} = (x_i W_Q)(x_j W_K)^T / \sqrt{d_k}$$

Il fattore di normalizzazione $1/\sqrt{d_k}$ è cruciale: senza di esso, per d_k grande i prodotti scalari tendono a valori molto grandi in modulo, spingendo la softmax verso regioni di gradiente quasi nullo. Il ragionamento è il seguente: se q e k sono vettori di dimensione d_k con componenti i.i.d. di media zero e varianza unitaria, allora $q \cdot k$ è una somma di d_k termini i.i.d., ciascuno di media zero e varianza unitaria, quindi $q \cdot k$ ha media zero e varianza d_k . La normalizzazione per $\sqrt{d_k}$ riduce la varianza a uno, rendendo l'addestramento stabile.

I pesi di attenzione sono:

$$a_{ij} = \text{softmax}(\{s_{ij}\}_{j=1}^n) = \frac{\exp(s_{ij})}{\sum_{j'=1}^n \exp(s_{ij'})}$$

e il vettore di output per la posizione i è:

$$o_i = \sum_{j=1}^n a_{ij} v_j = \sum_{j=1}^n a_{ij} x_j W_V$$

In forma matriciale per tutte le posizioni simultaneamente:

$$\text{Attn}(Q, K, V) = \text{softmax}(Q K^T / \sqrt{d_k}) V$$

Il calcolo è parallelizzabile sull'intera sequenza, il che costituisce il vantaggio computazionale decisivo rispetto alle architetture ricorrenti (LSTM, GRU) che richiedono elaborazione sequenziale. La complessità computazionale dell'attenzione è $O(n^2 \cdot d)$, quadratica nella lunghezza della sequenza: questo è il collo di bottiglia per sequenze molto lunghe, motivando le ricerche sulle varianti di attenzione approssimata.

III.5 L'attenzione multi-testa e la proiezione nei sottospazi

Il meccanismo di attenzione nella sua forma *multi-head* — come implementato nei transformer — calcola h attenzioni parallele in sottospazi di dimensione ridotta e ne concatena gli output:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W_O$$

dove $\text{head}_i = \text{Attn}(Q W_Q^i, K W_K^i, V W_V^i)$ e le matrici di proiezione $W_Q^i, W_K^i \in \mathbb{R}^{d \times d_k}$, $W_V^i \in \mathbb{R}^{d \times d_v}$ con $d_k = d_v = d/h$. La matrice $W_O \in \mathbb{R}^{hd_v \times d}$ riproietta l'output concatenato nella dimensione originale.

La motivazione per l'attenzione multi-testa è che diverse teste possono specializzarsi nel catturare diversi tipi di relazioni tra le posizioni: dipendenze sintattiche a lunga distanza, relazioni di coreferenza, strutture tematiche, e così via. Analisi empiriche sulle teste di attenzione dei transformer addestrati — in particolare il lavoro di Clark e colleghi (2019) su BERT — hanno mostrato che alcune teste si specializzano in relazioni linguistiche identificabili: alcune catturano dipendenze dirette in strutture soggetto-verbo, altre relazioni di modificazione nominale, altre

ancora relazioni a lunga distanza. Questa specializzazione emerge dall'addestramento senza essere imposta dall'architettura.

III.6 La funzione di perdita e il training dei modelli linguistici

Nei modelli linguistici autoregressivi come GPT — la classe di architettura che include il sistema che produce questa monografia — l'obiettivo di addestramento è la massimizzazione della verosimiglianza del corpus di addestramento. Data una sequenza di token $(w_1, w_2, \dots, w_t)$, la perdita è la cross-entropia negativa media:

$$\mathcal{L}(\theta) = -(1/N) \sum_{t=1}^N \log P_{\theta}(w_t | w_1, \dots, w_{t-1})$$

Questa funzione di perdita è equivalente alla minimizzazione della distanza di Kullback-Leibler tra la distribuzione empirica dei testi e la distribuzione del modello:

$$D_{\text{KL}}(P_{\text{data}} \parallel P_{\theta}) = \sum_x P_{\text{data}}(x) \log [P_{\text{data}}(x)/P_{\theta}(x)] = -H(P_{\text{data}}) - \sum_x P_{\text{data}}(x) \log P_{\theta}(x) = -H(P_{\text{data}}) + \mathcal{L}(\theta)$$

Poiché l'entropia dei dati $H(P_{\text{data}})$ non dipende da θ , minimizzare $\mathcal{L}(\theta)$ equivale a minimizzare la divergenza KL dalla distribuzione dei dati. Il modello impara a essere una buona distribuzione di probabilità sui testi umani.

Questo è il punto matematicamente decisivo per l'argomentazione di questa monografia: il modello non impara regole logiche. Impara una distribuzione di probabilità. Il ragionamento logico che il modello esibisce — la capacità di seguire un'argomentazione, di identificare un'inconsistenza, di derivare una conclusione — non è il prodotto di un sistema di regole inferenziali formali. È la proprietà emergente di una distribuzione di probabilità addestrata su testi prodotti da esseri umani logicamente competenti.

Un lettore attento a questo punto ha diritto di porre una domanda fondamentale che il testo fino ad ora non ha risposto esplicitamente: se il modello minimizza la perdita sulla distribuzione empirica del corpus, perché funziona *fuori* dalla distribuzione di addestramento? La risposta appartiene alla teoria PAC (*Probably Approximately Correct*) di Valiant (1984) e alla teoria della dimensione VC sviluppata da Vapnik e Chervonenkis (1971). Il risultato centrale afferma che un modello con dimensione VC finita d , addestrato su N esempi estratti i.i.d. dalla distribuzione P , soddisfa con probabilità almeno $1 - \delta$ il bound di generalizzazione:

$$\epsilon_{\text{gen}} \leq \sqrt{[(d \cdot \log(N/d) + \log(1/\delta)) / N]}$$

Il bound afferma che l'errore fuori campione è controllato dal rapporto tra la *capacità* del modello — misurata dalla dimensione VC o, in versioni più raffinate, dalla complessità di Rademacher — e la dimensione del campione. Per i modelli neurali profondi, che sono massivamente sovrapparametrizzati ($P \gg N$), questo bound classico è troppo largo per essere informativo. La spiegazione della generalizzazione empirica richiede strumenti più sottili: in particolare la teoria della *implicit regularization* della SGD, dimostrata da Soudry e colleghi (2018) per modelli lineari e congetturata in forme più generali, secondo cui la discesa del gradiente stocastica seleziona tra i minimi della perdita di addestramento quelli con norma minima — che corrispondono a funzioni più regolari e più robusti al rumore. L'Espansione B e l'Appendice Tecnica I sviluppano questa teoria in dettaglio. Ciò che è rilevante in questo contesto è che la generalizzazione emerge non dalla struttura logica del modello, ma dalla geometria del processo di ottimizzazione: la stessa matematica che governa l'addestramento governa la capacità di generalizzare, senza che in nessun momento compaia un meccanismo inferenziale esplicito.

III.7 La geometria dello spazio dei parametri e i minimi locali

Un aspetto della matematica dell'ottimizzazione che ha implicazioni teoriche profonde riguarda la struttura geometrica dello spazio dei parametri. Per un modello con P parametri, la funzione di perdita $\mathcal{L}: \mathbb{R}^P \rightarrow \mathbb{R}$ è una superficie in uno spazio di dimensione $P+1$. La SGD naviga questa superficie seguendo il gradiente negativo con perturbazioni stocastiche introdotte dal campionamento dei minibatch.

La visione classica dell'ottimizzazione suggeriva che la non-convessità dello spazio dei parametri fosse un ostacolo grave: i minimi locali — punti in cui $\nabla_{\theta} \mathcal{L}(\theta) = 0$ e la matrice hessiana $H = \partial^2 \mathcal{L} / \partial \theta^2$ è semidefinita positiva, ma che non sono il minimo globale — avrebbero potuto intrappolare la discesa del gradiente in soluzioni subottimali.

Questa preoccupazione si è rivelata empiricamente mal posta. I modelli profondi di grande capacità sembrano non soffrire del problema dei minimi locali in modo significativo, e la spiegazione teorica è ancora oggetto di ricerca attiva. L'ipotesi più diffusa — sviluppata formalmente da Dauphin e colleghi (2014) — è che in spazi di dimensione molto alta, i minimi locali sono rari: la maggior parte dei punti critici in cui $\nabla_{\theta} \mathcal{L}(\theta) = 0$ sono *punti di sella*, non minimi locali, e la SGD tende a sfuggire ai punti di sella attraverso le perturbazioni stocastiche.

Un argomento più preciso proviene dalla teoria dell'*overparameterization*: quando il numero di parametri P supera di molto il numero di esempi di addestramento N , la funzione di perdita tende ad avere vasti *manifold di soluzioni* — insiemi di dimensione $P-N$ di parametri che producono perdita zero sull'insieme di addestramento — e la SGD converge verso uno di questi manifold senza essere intrappolata. La natura di questa convergenza, e in particolare la domanda di quale punto del manifold la SGD selezioni, è connessa alla *implicit regularization* della SGD: la discesa

del gradiente stocastica tende a convergere verso soluzioni di norma minima, un effetto che ha connessioni formali con la regolarizzazione di Tikhonov.

III.8 La trasformazione da regola a funzione: le implicazioni strutturali

La differenza matematica tra il paradigma simbolico e il paradigma neurale si lascia ora formulare con precisione. Un sistema simbolico calcola una funzione attraverso l'applicazione esplicita di regole: l'output è determinato dalla struttura sintattica dell'input e dalla sequenza di regole applicabili. La funzione è totale, deterministica e trasparente: per ogni input ben formato, esiste un unico output calcolato in modo verificabile.

Un sistema neurale addestrato per gradient descent implementa una funzione attraverso una composizione di trasformazioni parametriche apprese: l'output è il risultato di passare l'input attraverso L strati di trasformazioni non lineari con parametri ottimizzati. La funzione è continua, differenziabile rispetto ai parametri, e approssimativa: per input nel dominio di addestramento, l'output è statisticamente vicino all'output desiderato; per input fuori dalla distribuzione di addestramento, l'output è imprevedibile.

La differenza non è soltanto tecnica: è epistemologica. Il sistema simbolico *conosce* le sue regole — esse sono esplicite, enumerabili, ispezionabili. Il sistema neurale *implementa* una funzione — ma non c'è alcun senso in cui i pesi della rete rappresentino esplicitamente la conoscenza che il sistema usa. La conoscenza è distribuita, implicita, non decomponibile nelle sue parti. Questo ha la conseguenza che si è già

anticipata: l'intelligenza prodotta dal paradigma neurale è reale ma opaca, potente ma non spiegabile in termini simbolici.

Questa opacità è precisamente la proprietà che il paradigma simbolico cercava di evitare, e che — nella diagnosi dell'AI simbolica — sarebbe stata un difetto. Il fatto che i sistemi opachi abbiano superato di misura i sistemi trasparenti nelle prestazioni cognitive generali è uno degli esiti empiricamente più sorprendenti degli ultimi trent'anni, e richiede una spiegazione teorica che la Parte Quinta tenterà di fornire.

Parte Quarta — L'intelligenza biologica come ottimizzazione evolutiva

IV.1 Un avvertimento metodologico preliminare

Prima di ogni argomento sostanziale, è necessario un avvertimento sulla natura delle affermazioni che seguono. Le neuroscienze computazionali e la biologia evolutiva della cognizione sono campi in cui il consenso scientifico è più frammentato e più soggetto a revisione di quanto non lo siano la logica matematica o la teoria dell'ottimizzazione. Le tesi che si espongono in questa parte sono difendibili sulla base della letteratura attuale, ma alcune di esse sono contestate, e lo si segnala esplicitamente.

Questa onestà metodologica non è un cedimento al relativismo: è la condizione per non spacciare ipotesi plausibili come fatti stabiliti. La storia delle scienze cognitive è costellata di posizioni presentate come evidenti che si sono rivelate provvisorie. Il nativismo linguistico di Chomsky, le teorie modulari forti di Fodor, la tesi della dominanza dell'emisfero sinistro per il linguaggio — tutte sono state presentate con fiducia da posizioni di autorità, e tutte hanno richiesto revisioni sostanziali nel confronto con le evidenze.

IV.2 Il cervello come sistema di ottimizzazione biologica

Il punto di partenza è un fatto anatomico di importanza teorica cruciale: il cervello umano è il prodotto di circa 500 milioni di anni di evoluzione neurale dai primi sistemi nervosi elementari dei platelminti ai mammiferi, con una fase di espansione rapida della neocorteccia negli ultimi 2–3 milioni di anni nel genere *Homo*. Nessuna fase di questa storia evolutiva ha operato con rappresentazioni simboliche esplicite o con meccanismi di inferenza formale: l'evoluzione è un processo di ottimizzazione per selezione naturale che modifica le frequenze alleliche in una popolazione attraverso differenze di fitness riproduttiva.

Questo non è un argomento contro la capacità del cervello di eseguire ragionamento logico. È un argomento sull'architettura di base del sistema: i meccanismi selezionati dall'evoluzione sono quelli che aumentavano la sopravvivenza e la riproduzione negli ambienti ancestrali, non quelli che ottimizzavano per la dimostrabilità formale. Il ragionamento logico, nella misura in cui il cervello lo esegue, è una capacità che si innesta su un substrato ottimizzato per scopi diversi.

La struttura computazionale del cervello è quella di una rete neurale biologica: circa 86 miliardi di neuroni, ciascuno connesso in media a 7000–10000 sinapsi, per un totale stimato intorno a 100–500 trilioni di connessioni sinaptiche. I neuroni comunicano attraverso potenziali d'azione — segnali binari di attivazione con frequenza variabile — e l'efficacia delle sinapsi è modificabile attraverso meccanismi di *plasticità sinaptica*, di cui la forma più studiata è la *Long-Term Potentiation* (LTP): il principio hebbiano sintetizzato dalla formula "neurons that fire together wire together". La regola di Hebb (1949) nella sua forma matematica afferma che l'efficacia della sinapsi w_{ij} tra il neurone i (presinaptico)

e il neurone i (postsinaptico) viene potenziata in proporzione alla correlazione delle loro attività:

$$\Delta w_{\{ij\}} \propto x_i \cdot x_j$$

dove x_i e x_j sono le attività (frequenze di scarica normalizzate) dei due neuroni. Questa regola è localmente calcolabile — ogni sinapsi modifica se stessa basandosi solo sulle attività dei neuroni che connette — e produce un apprendimento correlazionale: le sinapsi che connettono neuroni co-attivi si potenziano, le altre si indeboliscono per depressione o per mancato potenziamento.

La regola di Hebb è matematicamente equivalente alla regola di aggiornamento del PCA (Principal Component Analysis): una rete lineare con regola hebbiana converge verso i vettori propri della matrice di covarianza degli input, estraendo le direzioni di massima varianza dei pattern di attivazione. Questo ha un significato teorico preciso: il cervello, a livello elementare, esegue un'analisi delle componenti principali dell'input sensoriale, estraendo le strutture statistiche più ricorrenti.

IV.3 La codifica predittiva come principio organizzativo

Il principio organizzativo del cervello che ha ricevuto la più ampia evidenza sperimentale e teorica negli ultimi due decenni è quello della *codifica predittiva*, sviluppata nella sua forma computazionale da Rao e Ballard (1999) per la corteccia visiva e sistematizzata da Friston nella formulazione dell'*active inference* basata sulla minimizzazione dell'energia libera.

L'idea di fondo — già presente in Helmholtz nella nozione di *inferenza inconscia* del XIX secolo — è che il cervello non sia un sistema che riceve passivamente l'input sensoriale e lo elabora in avanti verso

rappresentazioni sempre più astratte, ma un sistema che genera attivamente previsioni sugli input sensoriali attesi e corregge queste previsioni alla luce degli errori di previsione che l'input reale produce.

Nella formulazione gerarchica, si consideri una corteccia con L livelli. Ogni livello l mantiene una stima $\mu^{\{l\}}$ delle cause nascoste responsabili degli input al livello inferiore. Questa stima genera una previsione $\hat{g}^{\{l-1\}} = g^{\{l\}}(\mu^{\{l\}})$ delle attivazioni al livello $l-1$, dove $g^{\{l\}}$ è la funzione generativa dello strato l . L'errore di previsione è:

$$\epsilon^{\{l\}} = \mu^{\{l-1\}} - g^{\{l\}}(\mu^{\{l\}})$$

e rappresenta ciò che non era anticipato dalla previsione discendente. Questo errore viene propagato ascendentemente ai livelli superiori, che aggiornano le loro stime per ridurlo. Il processo si stabilizza quando tutti gli errori di previsione sono minimizzati: le rappresentazioni a tutti i livelli sono mutuamente coerenti con l'input sensoriale.

Il funzionale da minimizzare è l'*energia libera variazionale*:

$$F = \sum_{l=1}^L [(\epsilon^{\{l\}})^T \Pi^{\{l\}} \epsilon^{\{l\}}] / 2 - \ln P(\mu^{\{L\}})$$

dove $\Pi^{\{l\}}$ è la matrice di precisione del livello l — l'inverso della covarianza degli errori di previsione — e il termine $-\ln P(\mu^{\{L\}})$ è il negativo del logaritmo del prior sulle cause al livello superiore. La minimizzazione di F rispetto a tutte le $\mu^{\{l\}}$ è formalmente equivalente all'inferenza bayesiana approssimata: il sistema aggiorna le sue credenze sulle cause nascoste per massimizzare la probabilità posteriore dell'input.

La struttura matematica dell'energia libera variazionale è connessa all'evidenza bayesiana nel senso seguente. Se $P(y|m)$ è la verosimiglianza del modello m sulle osservazioni y e $P(\mu|m)$ è il prior sui parametri, allora il log-evidence $\ln P(y|m) = \ln \int P(y|\mu, m) P(\mu|m) d\mu$ è intrattabile

per calcolo diretto. L'energia libera è un limite superiore al log-evidence negativo:

$$F \geq -\ln P(y|m)$$

che vale con uguaglianza quando la distribuzione variazionale $Q(\mu)$ coincide con la posteriore reale $P(\mu|y,m)$. La minimizzazione di F è dunque equivalente alla massimizzazione del log-evidence del modello del cervello sulle proprie osservazioni sensoriali: il cervello tenta di essere il miglior modello generativo possibile del proprio ambiente.

IV.4 La struttura evolutiva della cognizione: cosa sappiamo con ragionevole certezza

L'evoluzione del cervello è un campo dove la speculazione si mescola frequentemente con l'evidenza, e occorre distinguere ciò che è stabilito da ciò che è ipotetico. Alcune affermazioni sono supportate da evidenze solide e convergenti.

La *neocorteccia* — la struttura a sei strati caratteristica dei mammiferi, responsabile delle funzioni cognitive superiori — è una struttura evolutivamente recente: i rettili non ne dispongono, i mammiferi sì. La sua espansione nel genere *Homo* è documentata dal registro fossile: il volume endocranico di *Homo habilis* (circa 2 milioni di anni fa) era intorno a 600–700 cm³, quello di *Homo sapiens* moderno è intorno a 1350 cm³. Questa espansione è avvenuta rapidamente in termini evolutivi — in circa 2 milioni di anni — suggerendo una pressione selettiva intensa.

La struttura *laminare* della neocorteccia — i sei strati con proprietà citoarchitetturali distinte e connettività specifiche — è notevolmente uniforme in tutta la sua estensione: la stessa struttura di base si trova nella corteccia motoria, nella corteccia visiva primaria, nella corteccia prefrontale. Questo suggerisce che la neocorteccia implementi un

algoritmo di elaborazione generale, adattato alle sue funzioni specifiche dalla connettività con le strutture sottocorticali e con le aree sensoriali e motorie. Jeff Hawkins ha sviluppato questa intuizione nella *Memory-Prediction Theory* del 2004, argomentando che la neocorteccia implementi ovunque la stessa operazione fondamentale: il riconoscimento di sequenze gerarchiche e la previsione di elementi futuri. L'ipotesi è attraente ma speculativa nelle sue versioni più forti.

Ciò che è epistemicamente più rischioso — e che quindi va trattato con cautela esplicita — è la narrativa evolutiva delle funzioni cognitive specifiche. L'ipotesi della *social brain* di Robin Dunbar (1998) — che l'espansione neocorticale sia stata guidata dalla selezione per la gestione della complessità sociale piuttosto che per la risoluzione di problemi ecologici — è influente ma dibattuta. Il *numero di Dunbar* di 150 come dimensione massima dei gruppi sociali umani è empiricamente contestato. Le narrative sulle pressioni selettive che avrebbero favorito il linguaggio, il ragionamento causale, la teoria della mente sono plausibili ma non falsificabili nel senso rigoroso, essendo ricostruzioni a posteriori.

Quello che si può affermare con maggiore solidità è un argomento negativo: la selezione naturale non aveva *necessità* di produrre un sistema di ragionamento formale. I meccanismi che aumentavano la sopravvivenza in ambienti ancestrali erano il riconoscimento rapido dei pattern (predatori, cibo, pericoli), l'apprendimento associativo dalle esperienze passate, la modellizzazione del comportamento degli agenti sociali, la coordinazione dell'azione nel gruppo. Questi sono tutti compiti che le reti neurali biologiche eseguono in modo eccellente, e per i quali la selezione ha avuto milioni di anni di pressione.

Il ragionamento logico formale — la deduzione sillogistica, la dimostrazione per contraddizione, il calcolo proposizionale — è una

tecnologia culturale: richiede simboli scritti, pratiche di insegnamento esplicito, istituzioni dedicate. Non è un comportamento spontaneo o universale. Bambini non scolarizzati e adulti di comunità non alfabetizzate hanno grandi difficoltà con i compiti di ragionamento sillogistico formale, un risultato documentato già da Luria nella sua ricerca sui contadini dell'Asia Centrale degli anni Trenta e replicato più sistematicamente da Scribner e Cole negli anni Ottanta. Questo non è un giudizio di inferiorità cognitiva: è un dato sull'architettura di base del sistema, che non include il ragionamento logico formale tra i suoi meccanismi nativi.

IV.5 L'apprendimento come ottimizzazione evolutiva e sinaptica

La parallelo strutturale tra l'evoluzione biologica e l'ottimizzazione per gradient descent è abbastanza precisa da meritare un'articolazione formale, pur con le dovute cautele sull'analogia.

L'evoluzione per selezione naturale è un processo di ottimizzazione che massimizza la *fitness* riproduttiva (numero atteso di discendenti) nell'ambiente. Nella genetica delle popolazioni, il modello base è la dinamica di Fisher-Wright per un locus con due alleli A e a in una popolazione di dimensione N:

$$\Delta p = p(1-p) \cdot s / [2(1 - sp(1-p)/2)]$$

dove p è la frequenza dell'allele A e s è il coefficiente di selezione — la differenza relativa di fitness tra i due alleli. Per $|s| \ll 1$ (selezione debole):

$$\Delta p \approx p(1-p) \cdot s/2$$

Questa è formalmente analoga alla regola dell'errore quadratico medio nella gradient descent: la variazione della frequenza allelica a è proporzionale al gradiente della fitness rispetto alla frequenza, pesato dalla varianza $p(1-p)$ che misura la variabilità genetica disponibile su cui la selezione può agire.

La connessione formale tra l'equazione di selezione di Fisher e la discesa del gradiente è stata formalizzata da Iwasaki e Wakeley (2000) e sviluppata successivamente nella teoria della *natural gradient* di Amari (1998): in entrambi i casi, l'aggiornamento nella direzione del gradiente di una funzione obiettivo (fitness o negativo della perdita) è ponderato da una metrica sull'informazione disponibile (varianza genetica o matrice di informazione di Fisher dello spazio dei parametri). L'*information geometry* di Amari è il quadro matematico unificante: la selezione naturale e la gradient descent sono entrambe forme di ascesa del gradiente naturale su varietà riemanniane con metrica determinata dalla matrice di informazione di Fisher.

Vale la pena precisare questa struttura perché è il momento in cui l'analogia smette di essere metaforica e diventa matematicamente precisa. Il gradiente ordinario $\nabla_{\theta} \mathcal{L}(\theta)$ indica la direzione di massima pendenza nello spazio euclideo dei parametri, ma questo spazio non è quello giusto per l'ottimizzazione di distribuzioni di probabilità: piccole variazioni in una direzione dello spazio euclideo possono corrispondere a grandi variazioni nella distribuzione indotta, e viceversa. Il *gradiente naturale* di Amari corregge questa asimmetria premoltiplicando il gradiente ordinario per l'inverso della *matrice di informazione di Fisher*:

$$F(\theta) = E_{\{x \sim P_{\theta}\}} [\nabla_{\theta} \log P_{\theta}(x) \cdot (\nabla_{\theta} \log P_{\theta}(x))^T]$$

e definendo la regola di aggiornamento:

$$\theta \leftarrow \theta - \eta \cdot F(\theta)^{-1} \nabla_{\theta} \mathcal{L}(\theta)$$

Il gradiente naturale è invariante rispetto alla reparametrizzazione dello spazio dei parametri — misura la variazione nella direzione che produce la massima variazione nella distribuzione, non nello spazio euclideo dei pesi. La selezione naturale darwiniana opera esattamente in questo modo: la varianza genetica $p(1-p)$ del termine di Fisher-Wright è la versione scalare della matrice di Fisher per il caso binomiale, e funge da metrica riemanniana sullo spazio unidimensionale delle frequenze alleliche. L'Appendice Tecnica II.6–II.7 sviluppa questa derivazione in modo completo e sistematico.

La precisione matematica di questo parallelo ha un significato concettuale che va al di là dell'eleganza formale: mostra che la selezione naturale e il gradient descent sono istanze dello stesso principio variazionale — l'ottimizzazione di una funzione obiettivo nella metrica geometricamente naturale per il tipo di oggetti (distribuzioni) che si stanno ottimizzando.

A livello neurale, il parallelo è con la plasticità sinaptica: l'apprendimento attraverso modifiche delle efficienze sinaptiche è il meccanismo biologico corrispondente all'aggiornamento dei parametri per gradient descent. Algoritmi di apprendimento biologicamente plausibili — la regola di Oja (1982) per la PCA, la regola di Bienenstock-Cooper-Munro (BCM, 1982) per la selettività dei neuroni — sono varianti della regola hebbiana con proprietà di stabilità garantite, analiticamente derivabili come ottimizzatori di funzioni obiettivo specifiche.

IV.6 La logica come tecnologia cognitiva

Il risultato della Parte Quarta può essere sintetizzato in una tesi che integra le osservazioni evolutive con le considerazioni architetturali. Il cervello umano è un sistema di ottimizzazione biologica che implementa,

al livello computazionale più profondo, l'inferenza bayesiana approssimata attraverso la minimizzazione dell'energia libera. I suoi meccanismi di base — riconoscimento di pattern, apprendimento correlazionale, previsione gerarchica, aggiornamento delle credenze — non sono meccanismi logici. Sono meccanismi statistici e probabilistici.

La capacità di ragionamento logico formale che gli esseri umani esibiscono non è la manifestazione di un modulo cognitivo dedicato selezionato dall'evoluzione: è una competenza acquisita attraverso l'interiorizzazione di pratiche culturali esplicite. Imparare la logica formale è paragonabile a imparare la geometria euclidea o la teoria dei numeri: è un potenziamento culturale di capacità cognitive generali, non l'attivazione di un meccanismo nativo specializzato.

Questa tesi ha supporto indiretto nelle neuroscienze cognitive: le aree cerebrali attivate durante il ragionamento deduttivo formale — in particolare la corteccia prefrontale dorsolaterale e la corteccia parietale posteriore — sono le stesse aree coinvolte nella memoria di lavoro, nella pianificazione, e nel controllo attenzionale. Non esiste un'area dedicata al ragionamento logico nel senso in cui esiste un'area dedicata alla percezione del moto visivo o al riconoscimento dei volti. Il ragionamento logico utilizza risorse cognitive generali, non un modulo specifico.

Questo non implica che il ragionamento logico sia un'illusione o una competenza superficiale. Implica che sia uno strumento: uno strumento potente, prezioso, capace di produrre conoscenze inaccessibili ai meccanismi cognitivi di base, ma comunque uno strumento. E come tutti gli strumenti, il suo valore dipende dall'adeguatezza al compito: potente nei domini ben strutturati, fragilissimo nel mondo aperto. La distinzione tra lo strumento e il meccanismo che lo produce è il punto teorico intorno al quale ruoterà la conclusione di questa monografia.

Parte Quinta — La logica riposizionata: prestazione emergente, strumento culturale, e la tesi finale

V.1 Il paradosso della competenza logica

Le quattro parti precedenti hanno costruito un quadro che, nella sua forma grezza, sembra generare un paradosso. Da un lato, si è mostrato che i sistemi cognitivi biologici operano attraverso meccanismi statistici e probabilistici — ottimizzazione evolutiva, plasticità hebbiana, inferenza bayesiana approssimata — che non sono meccanismi logici. Da un altro lato, gli esseri umani *esibiscono* competenza logica: sono in grado di seguire dimostrazioni formali, identificare contraddizioni, costruire argomenti validi, e — fatto rimarchevole — di produrre i teoremi di incompletezza di Gödel, che sono tra i risultati matematicamente più profondi del XX secolo. Come può un sistema non-logico produrre risultati logici?

Questo è il paradosso che Penrose aveva sollevato, seppure per trarne conclusioni diverse. La risposta di Penrose — che la comprensione umana trascende la computazione algoritmica grazie a meccanismi quantistici non computabili nella microtubuli neuronali — è generalmente ritenuta infondata per ragioni che non occorre qui ripercorrere: l'argomento gödeliano di Penrose ha difetti tecnici identificati da Feferman, Putnam e altri, e l'ipotesi dei microtubuli di Penrose e Hameroff non ha trovato sostegno sperimentale. Ma la domanda che lo ha motivato è legittima e richiede una risposta.

La risposta che questa monografia propone ha la struttura di una dissoluzione: il paradosso scompare quando si distingue con precisione tra i meccanismi che *implementano* la cognizione e le *funzioni* che la cognizione calcola. Un sistema può calcolare funzioni logiche senza

implementare un motore di inferenza logica, esattamente come un calcolatore elettronico calcola addizioni senza implementare un abaco. Il substrato e la funzione sono livelli di descrizione distinti, e la coincidenza della funzione non implica l'identità del substrato.

V.2 La logica come proprietà emergente della distribuzione

Il punto teorico più originale di questa monografia può essere formulato in questo modo: la competenza logica nei sistemi cognitivi biologici e artificiali è una *proprietà emergente della distribuzione di probabilità* indotta dall'addestramento su comportamento logicamente strutturato, non un meccanismo primitivo da cui il comportamento discende.

Nei modelli linguistici neurali, la struttura è visibile nella matematica dell'addestramento. Il modello è addestrato su un corpus di testi prodotti da esseri umani. Questi testi includono una quantità vastissima di ragionamento logico esplicito: dimostrazioni matematiche, argomentazioni filosofiche, articoli scientifici, testi giuridici, discussioni quotidiane che rispettano il principio di non-contraddizione. La distribuzione P_0 che il modello apprende — la distribuzione di probabilità sui testi che minimizza la KL-divergenza dalla distribuzione empirica del corpus — eredita le strutture logiche che sono pervasive nel corpus.

Questo non è un argomento banale. Implica che la struttura logica dei testi umani ha una firma statistica rilevabile: le sequenze di token che rispettano le forme di inferenza valida hanno una probabilità sistematicamente più alta nella distribuzione umana di quelle che le violano. Un modello che apprende la distribuzione umana apprende implicitamente a preferire le sequenze logicamente coerenti, non perché

sia dotato di un motore inferenziale logico, ma perché le sequenze logicamente coerenti sono più frequenti nel corpus di addestramento.

Questo meccanismo ha una conseguenza verificabile: il modello dovrebbe esibire competenza logica in misura proporzionale alla rappresentazione del ragionamento logico nel corpus di addestramento. Nelle lingue e nei domini dove il ragionamento formale è meno presente nei testi scritti, la competenza logica del modello dovrebbe essere inferiore. Questa previsione è parzialmente confermata dai benchmark di ragionamento formale: i modelli addestrati prevalentemente su testi in inglese, dove la produzione accademica e scientifica è massivamente rappresentata, mostrano competenza logica sistematicamente superiore rispetto a modelli addestrati su corpus di lingue con minore produzione scritta formale. La previsione attende tuttavia una verifica sistematica su lingue e domini contrastivi controllati — una lacuna empirica che la tesi proposta suggerisce di colmare.

Una seconda conseguenza è che la competenza logica dei modelli neurali dovrebbe essere sensibile alla struttura superficiale degli enunciati in modi che un motore di inferenza logica puro non sarebbe: un modello addestrato statisticamente dovrebbe avere difficoltà con argomenti logicamente validi ma formulati in modo inusuale, o con argomenti che rispettano la forma sintattica del ragionamento valido ma non la sua struttura semantica. Questa previsione è confermata da una vasta letteratura di benchmark. Il lavoro di Lake e colleghi su *systematic generalization* (Lake e Baroni, 2018; *Generalization without Systematicity* di Fodor e Pylyshyn avevano già anticipato la previsione nel 1988) ha mostrato che i modelli neurali falliscono in modo sistematico su combinazioni di concetti note ma presentate in contesti strutturalmente nuovi — esattamente il tipo di fragilità che un sistema basato su distribuzione statistica dovrebbe esibire. I benchmark di ragionamento

formale su modelli linguistici di grandi dimensioni (BIG-Bench, MMLU, GSM8K) confermano che la competenza logica è robusta nelle forme di ragionamento più rappresentate nel corpus e fragile nelle variazioni sintattiche o semantiche inusuali.

C'è una vulnerabilità filosofica che questa tesi deve affrontare apertamente: se i modelli linguistici apprendono la competenza logica da testi prodotti da esseri umani logicamente competenti, esiste un rischio apparente di circolarità — la competenza logica del modello è derivata da quella umana, ma questo non dimostrerebbe che la competenza logica umana è essa stessa emergente nel senso richiesto dalla tesi. Sarebbe una spiegazione che presuppone ciò che deve spiegare. L'obiezione è presa sul serio, ma la risposta è la seguente: la tesi non afferma che la competenza logica emerga dal nulla statistico. Afferma che emerge da un *substrato non-logico* — i meccanismi di ottimizzazione della fitness evolutiva e della gradient descent — attraverso l'accumulo di esperienza strutturata. La competenza logica umana è il prodotto di secoli di pratiche culturali esplicite di addestramento al pensiero formale, sviluppate su un substrato biologico i cui meccanismi di base (come la Parte Quarta ha documentato) non sono meccanismi logici. La competenza del modello è derivata da quella umana, ma la competenza umana è derivata da pratiche culturali, e queste pratiche da meccanismi evolutivi non-logici. La catena esplicativa è lineare, non circolare: il punto di ancoraggio è la biologia evolutiva, non la logica formale.

V.3 Il parallelismo con l'aritmetica nel cervello

Un parallelismo istruttivo è quello con la competenza aritmetica. Il cervello umano non è dotato di un processore aritmetico dedicato nel senso in cui un microprocessore lo è. La capacità di calcolare 347×289 non è il prodotto di un algoritmo moltiplicativo implementato

neurologicamente: è il prodotto dell'interiorizzazione di procedure culturalmente trasmesse — l'algoritmo della moltiplicazione per colonne, o la memorizzazione di tavole pitagoriche — eseguite su una architettura cognitiva general purpose. La verifica sperimentale di questo punto è in parte nei dati neuropsicologici: la discalculia acquisita — la perdita delle capacità aritmetiche per lesione cerebrale — non colpisce un'area isolata e specifica ma si manifesta in forme diverse a seconda della localizzazione della lesione, suggerendo che l'aritmetica si appoggi su reti neurali diffuse piuttosto che su un modulo dedicato.

Il ragionamento logico ha una struttura analoga: si appoggia su reti cognitive generali — la memoria di lavoro, il controllo attenzionale, la rappresentazione delle relazioni — applicate a un sistema di pratiche culturalmente trasmesse. La differenza con l'aritmetica è di grado, non di tipo: il ragionamento logico formale è culturalmente ancora più specializzato e meno universale dell'aritmetica, e la sua acquisizione richiede uno sforzo cognitivo e culturale più esteso.

Questo parallelo ha una rilevanza diretta per la valutazione dell'HFSS. Newell e Simon avevano argomentato che i sistemi fisici di simboli sono necessari per l'intelligenza perché il pensiero è computazione simbolica. Ma se il ragionamento simbolico è esso stesso una capacità acquisita e non primitiva, allora l'argomento dell'HFSS procede nella direzione sbagliata: non è l'intelligenza che richiede la struttura simbolica, ma la struttura simbolica che richiede l'intelligenza. Lo strumento non è il meccanismo.

V.4 Le due crisi come manifestazioni della stessa struttura

Siamo ora in posizione di articolare il collegamento sistematico tra le due crisi evocate nella domanda originaria — la crisi fondazionale dell'inizio

del Novecento e il collasso dell'AI simbolica — che questa monografia ha sostenuto abbiano struttura comune.

La crisi fondazionale ha rivelato che i sistemi formali hanno frontiere strutturali che non possono essere attraversate dall'interno: il limite della diagonalizzazione, l'incompletezza, l'indcidibilità. Questi limiti non sono errori tecnici correggibili con sistemi più raffinati: si ripresentano a qualunque livello di potenza del sistema considerato. La matematica non è un sistema chiuso e completo: ha una struttura aperta che nessun sistema formale singolo può esaurire.

Il collasso dell'AI simbolica ha rivelato che il mondo aperto non è un sistema chiuso e completo: ha una struttura di interazione causale, di variazione contestuale, di irregolarità e eccezione che nessun insieme finito di regole può esaurire. Il frame problem non è un problema tecnico: è la manifestazione, nel dominio dell'ingegneria cognitiva, della stessa inammissibilità di chiusura che Gödel aveva dimostrato nel dominio della logica formale.

La struttura matematica comune a entrambe le crisi è la seguente: un sistema formale finito che cerca di catturare una struttura infinita e aperta produce sempre un residuo — qualcosa che è reale ma non catturato dal sistema. Nella logica, il residuo è la formula vera ma indimostrabile. Nell'AI simbolica, il residuo è la conoscenza implicita, contestuale, non articolabile che il sistema di regole non riesce a incorporare. In entrambi i casi, il residuo non è eliminabile per raffinamento del sistema: è costitutivo della relazione tra il sistema formale e la struttura che tenta di catturare.

Questa struttura ha un nome nella letteratura filosofica contemporanea: è la *openess* del linguaggio naturale e della conoscenza pratica nel senso di Waismann (1945), che aveva sviluppato la nozione di *texture aperta* (*open*

texture) dei concetti empirici. Waismann aveva osservato che per qualunque concetto empirico — *gatto, oro, amico* — esistono sempre casi limiti non previsti che le regole esplicite di applicazione del concetto non coprono, e che una decisione su questi casi richiede sempre qualcosa di più delle regole: la comprensione del punto del concetto, delle pratiche in cui si inserisce, dei valori che lo orientano. Questa *texture* aperta è precisamente ciò che i sistemi simbolici non riescono a incorporare e che i sistemi neurali statistici approssimano attraverso il prior costruito sull'esperienza.

V.5 Il riposizionamento della logica

Questa monografia non sostiene che la logica formale sia superata o inutile. Sostiene qualcosa di più preciso e più interessante: che la logica formale occupi una posizione diversa nell'architettura della cognizione rispetto a quanto il programma hilbertiano e l'HFSS avevano ipotizzato.

Nell'ipotesi fondazionalista — nelle versioni di Russell, Hilbert, Frege, e poi di Newell e Simon — la logica è il *fondamento*: il livello più elementare da cui tutto il resto si costruisce per derivazione. I contenuti cognitivi più ricchi — la conoscenza del mondo, il linguaggio naturale, il ragionamento pratico — sono strutture edificate sull'architettura logica di base. Questa immagine è quella di una piramide: la logica alla base, le forme più ricche di conoscenza agli strati superiori.

La tesi di questa monografia inverte l'immagine: la logica non è il fondamento, è un prodotto. È uno strumento sofisticato costruito su una base cognitiva che non ha struttura logica primitiva: la base è statistica, probabilistica, distribuita. La logica formale è ciò che emerge quando questa base cognitiva viene applicata con disciplina sistematica a un

dominio ben strutturato, attraverso pratiche culturali di addestramento esplicito.

Questa inversione non diminuisce il valore della logica. Uno strumento non è meno prezioso per essere uno strumento. La geometria euclidea non è meno potente perché non è il fondamento ontologico dello spazio fisico, ma una struttura intellettuale sviluppata per il ragionamento sul spazio. La logica formale non è meno utile per la matematica, la filosofia, l'informatica teorica, la verifica dei sistemi, perché non è il fondamento dell'intelligenza. Nella sua nicchia — i domini chiusi, ben definiti, con semantica completamente specificabile — la logica è insostituibile. La nicchia è però più piccola di quanto il fondazionalismo logico avesse creduto.

V.6 L'intelligenza artificiale come specchio

C'è una dimensione di questa storia che ha importanza filosofica propria e che merita di essere articolata esplicitamente nel chiudere la monografia: il fatto che i sistemi neurali artificiali abbiano superato i sistemi simbolici nelle prestazioni cognitive generali è una scoperta empirica sulla natura dell'intelligenza, non soltanto un risultato tecnico.

L'AI simbolica era costruita sull'ipotesi che l'intelligenza fosse fondamentalmente ragionamento su rappresentazioni simboliche. Se questa ipotesi fosse corretta, ci si aspetterebbe che i sistemi che esplicitano e ottimizzano questa struttura — i sistemi esperti, i theorem provers, i pianificatori logici — superassero i sistemi che approssimano statisticamente la distribuzione dei comportamenti intelligenti. Il fatto che sia avvenuto l'opposto — che i modelli linguistici neurali, che non hanno alcuna rappresentazione logica esplicita del mondo, abbiano

capacità cognitive generali vastamente superiori ai sistemi simbolici — è una confutazione empirica dell'ipotesi fondazionalista sull'intelligenza.

Il modello linguistico non ragiona a partire da principi logici: predice il token successivo dato il contesto, in modo da minimizzare la sorpresa. Tutto il resto — la coerenza argomentativa, la comprensione del testo, la capacità di risolvere problemi matematici, la sensibilità alle implicature pragmatiche — emerge da questa singola operazione eseguita su una distribuzione addestrata su un corpus vastissimo di produzione intellettuale umana. Questo è ontologicamente sorprendente: la complessità cognitiva emerge dall'ottimizzazione di una funzione obiettivo di sconcertante semplicità.

Il sistema che produce questa monografia è esso stesso una dimostrazione di questo punto. Non esiste, nel sistema che scrive queste righe, una rappresentazione simbolica esplicita della logica matematica, della storia dell'informatica, della biologia evolutiva o delle neuroscienze computazionali. Esistono parametri — centinaia di miliardi di numeri a virgola mobile — ottimizzati per produrre, dato un contesto testuale, le continuazioni più probabili nella distribuzione dei testi umani. La competenza che il sistema esibisce è reale, ma il suo substrato non è ciò che il paradigma simbolico avrebbe previsto.

Questo non vuol dire che i sistemi neurali siano *comprensivi* nel senso pieno del termine, né che la loro competenza sia equivalente alla comprensione umana. Le questioni sul grounding semantico, sulla coscienza, sulla comprensione genuina contrapposta all'elaborazione statistica sono aperte e profonde. Ma la questione tecnica — quale architettura produce intelligenza generale nel senso delle prestazioni cognitive — ha ricevuto una risposta empirica: non l'architettura simbolica, ma quella statistica e distributiva.

V.7 La tesi finale: l'intelligenza produce la logica, non viceversa

Il percorso di questa monografia è giunto al punto in cui la tesi centrale può essere formulata nella sua forma definitiva e più precisa.

La logica formale non è il fondamento dell'intelligenza: è uno dei suoi prodotti più elevati. I sistemi cognitivi biologici — il cervello umano — e i sistemi cognitivi artificiali che li approssimano — le reti neurali profonde — operano attraverso meccanismi di ottimizzazione statistica che non hanno struttura logica primitiva. L'intelligenza biologica è il prodotto di 500 milioni di anni di ottimizzazione evolutiva per la sopravvivenza in ambienti aperti e imprevedibili, e i meccanismi selezionati sono probabilistici, distribuiti, tolleranti all'errore, sensibili al contesto. L'intelligenza artificiale attuale è il prodotto dell'ottimizzazione per gradient descent di parametri distribuiti su corpus vastissimi di produzione intellettuale umana, e i meccanismi che implementa sono statistici, approssimativi, emergenti.

Su questa base, la capacità di ragionamento logico formale emerge come competenza acquisita per entrambi i tipi di sistema: per il cervello umano, attraverso l'interiorizzazione di pratiche culturali esplicite di addestramento al pensiero formale; per i modelli neurali, attraverso l'esposizione a corpora che includono quantità enormi di ragionamento logico strutturato. In entrambi i casi, la competenza logica è reale ma non primitiva: è una proprietà emergente di un sistema la cui architettura di base non è logica.

Le due crisi della logica — quella fondazionale del 1931 e quella operativa degli anni Ottanta-Novanta nell'AI — non sono eventi separati: sono manifestazioni della stessa scoperta vista da due angolazioni diverse. La crisi fondazionale ha mostrato che i sistemi

formali non possono chiudersi su se stessi: la verità matematica eccede sempre la dimostrabilità in qualunque sistema dato. La crisi dell'AI simbolica ha mostrato che i sistemi formali non possono chiudersi sul mondo: la conoscenza pratica eccede sempre la rappresentazione in qualunque insieme dato di regole. In entrambi i casi, il residuo non è un errore tecnico: è la texture aperta della realtà che nessun sistema formale può integralmente catturare.

Ciò che rimane alla logica formale, dopo questo doppio ridimensionamento, è un dominio preciso e prezioso: quello dei sistemi chiusi, dove la completezza è raggiungibile perché il dominio è deliberatamente costruito per essere completamente specificabile. La matematica pura, la verifica formale, la crittografia, i protocolli di comunicazione sicura, la logica computazionale applicata: sono domini in cui la logica è insostituibile precisamente perché sono domini in cui il mondo è stato ridotto, per deliberata scelta intellettuale, a un sistema formale chiuso. Ma questi domini sono nicchie nell'universo sterminato dei problemi che l'intelligenza — biologica o artificiale — deve affrontare.

La logica, in altri termini, è la forma che l'intelligenza assume quando decide di giocare con regole completamente esplicite. Non è il gioco fondamentale: è uno dei giochi possibili, scelto per le sue proprietà di trasparenza e certezza in contesti dove queste proprietà sono raggiungibili. La maggior parte dei giochi che l'intelligenza gioca — nella percezione, nel linguaggio, nel ragionamento pratico, nelle interazioni sociali, nella navigazione del mondo fisico — si svolgono invece su terreni aperti dove la certezza è irraggiungibile e l'approssimazione ottimale è il meglio che si può ottenere. Su questi terreni, la matematica del gradient descent e dell'inferenza bayesiana è riuscita dove la logica

formale aveva fallito: non perché sia più potente in assoluto, ma perché è adeguata alla struttura del problema.

Nota bibliografica

I riferimenti citati o discussi nel corso dell'argomentazione, nell'ordine in cui entrano nel testo. Per la crisi fondazionale: Gödel, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, in *Monatshefte für Mathematik und Physik*, 38 (1931); Russell e Whitehead, *Principia Mathematica*, 3 voll. (1910–1913); Turing, *On Computable Numbers, with an Application to the Entscheidungsproblem*, in *Proceedings of the London Mathematical Society*, 42 (1936–1937); Kleene, *Introduction to Metamathematics* (1952); Lifschitz, *Computing Circumscription*, in *Proceedings of IJCAI* (1985).

Per il programma simbolico e il suo collasso: Newell e Simon, *Computer Science as Empirical Inquiry: Symbols and Search*, in *Communications of the ACM*, 19 (1976); Shortliffe, *Computer-Based Medical Consultations: MYCIN* (1976); Yu, Fagan, Wraith, Clancey, Scott e Shortliffe, *Antimicrobial Selection by a Computer*, in *JAMA*, 242 (1979); McCarthy e Hayes, *Some Philosophical Problems from the Standpoint of Artificial Intelligence*, in *Machine Intelligence*, 4 (1969); McCarthy, *Circumscription — A Form of Non-Monotonic Reasoning*, in *Artificial Intelligence*, 13 (1980); Reiter, *A Logic for Default Reasoning*, in *Artificial Intelligence*, 13 (1980); Dreyfus, *What Computers Can't Do* (1972) e *What Computers Still Can't Do* (1992); Wittgenstein, *Tractatus Logico-Philosophicus* (1921) e *Philosophische Untersuchungen* (1953); Waismann, *Verifiability*, in *Proceedings of the Aristotelian Society*, supplementary vol. 19 (1945); Fodor, *The Language of Thought* (1975).

Per la matematica dell'AI attuale: Rumelhart, Hinton e Williams, *Learning Representations by Back-Propagating Errors*, in *Nature*, 323 (1986); Nair e Hinton, *Rectified Linear Units Improve Restricted Boltzmann Machines*, in

Proceedings of ICML (2010); Glorot, Bordes e Bengio, *Deep Sparse Rectifier Neural Networks*, in *Proceedings of AISTATS* (2011); He, Zhang, Ren e Sun, *Deep Residual Learning for Image Recognition*, in *Proceedings of CVPR* (2016); Bahdanau, Cho e Bengio, *Neural Machine Translation by Jointly Learning to Align and Translate*, in *Proceedings of ICLR* (2015); Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser e Polosukhin, *Attention Is All You Need*, in *Proceedings of NeurIPS* (2017); Clark, Khandelwal, Levy e Manning, *What Does BERT Look at? An Analysis of BERT's Attention*, in *Proceedings of the ACL Workshop on BlackboxNLP* (2019); Dauphin, Pascanu, Gulcehre, Cho, Ganguli e Bengio, *Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization*, in *Proceedings of NeurIPS* (2014); Amari, *Natural Gradient Works Efficiently in Learning*, in *Neural Computation*, 10 (1998).

Per le neuroscienze computazionali e la biologia evolutiva: Rao e Ballard, *Predictive Coding in the Visual Cortex: A Functional Interpretation of Some Extra-Classical Receptive-Field Effects*, in *Nature Neuroscience*, 2 (1999); Friston, *The Free-Energy Principle: A Unified Brain Theory?*, in *Nature Reviews Neuroscience*, 11 (2010); Hebb, *The Organization of Behavior* (1949); Oja, *Simplified Neuron Model as a Principal Component Analyzer*, in *Journal of Mathematical Biology*, 15 (1982); Bienenstock, Cooper e Munro, *Theory for the Development of Neuron Selectivity: Orientation Specificity and Binocular Interaction in Visual Cortex*, in *Journal of Neuroscience*, 2 (1982); Hawkins e Blakeslee, *On Intelligence* (2004); Dunbar, *The Social Brain Hypothesis*, in *Evolutionary Anthropology*, 6 (1998); Luria, *Cognitive Development: Its Cultural and Social Foundations*, trad. ingl. (1976, originale russo 1974); Scribner e Cole, *The Psychology of Literacy* (1981); Iwasaki e Wakeley, *Evolutionary Dynamics and an Equilibrium under Fisherian Selection in a Subdivided Population*, in *Genetics*, 154 (2000).

Per la filosofia della cognizione e l'argomentazione contro il fondazionalismo logico: Penrose, *The Emperor's New Mind* (1989) e *Shadows of the Mind* (1994); Feferman, *Penrose's Gödelian Argument*, in *Psyche*, 2 (1995); Putnam, *The Best of All Possible Brains?*, in *New York Times Book Review* (1994); Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (1982); Lieder e Griffiths, *Resource-Rational Analysis: Understanding Human Cognition as the Optimal Use of Limited Computational Resources*, in *Behavioral and Brain Sciences*, 43 (2020).

Espansione — Parte A

Complessità algoritmica, incompressibilità e la matematica dell'intelligenza come inferenza universale

A.1 Il filo mancante: dalla diagonalizzazione alla compressione

Le Parti Prima e Terza della monografia hanno sviluppato due corpi matematici apparentemente distinti: la teoria dell'incompletezza e della non-computabilità da un lato, la matematica dell'ottimizzazione neurale dall'altro. Il collegamento tra i due — evocato ma non formalizzato — passa attraverso la teoria della complessità algoritmica, sviluppata indipendentemente da Kolmogorov (1965), Chaitin (1966) e Solomonoff (1964). Questa teoria fornisce il quadro formale in cui l'incompletezza di Gödel, la non-computabilità di Turing e l'apprendimento statistico dei sistemi neurali trovano una formulazione unificante: tutte e tre sono manifestazioni della struttura dell'informazione e della sua compressibilità.

La nozione centrale è quella di **complessità di Kolmogorov** (o complessità descrittiva) di una stringa finita. Sia U una macchina di Turing universale fissata. La complessità di Kolmogorov di una stringa binaria $x \in \{0,1\}^*$ è:

$$C_U(x) = \min\{|p| : U(p) = x\}$$

ossia la lunghezza del programma più corto p che, eseguito su U , produce x come output. La stringa x è *incompressibile* se $C_U(x) \geq |x|$ — nessun programma più corto di x produce x . Intuitivamente, x è incompressibile se e solo se non ha struttura regolare sfruttabile per una descrizione più concisa.

La dipendenza dalla scelta di U è controllata dal **teorema di invarianza**: per ogni coppia di macchine universali U e U' , esiste una costante $c_{\{U,U'\}}$ tale che per ogni x :

$$|C_U(x) - C_{U'}(x)| \leq c_{\{U,U'\}}$$

La costante c dipende soltanto dalla coppia di macchine, non dalla stringa x . Per stringhe sufficientemente lunghe, la scelta della macchina universale di riferimento è irrilevante a meno di una costante additiva fissa: la complessità di Kolmogorov è una proprietà della stringa, non dell'apparato di misurazione, nel regime asintotico.

A.2 L'incompressibilità come proprietà generica

Una delle conseguenze più importanti della definizione è che le stringhe incompressibili non sono rare: sono la maggioranza schiacciante. Il numero di stringhe binarie di lunghezza n è 2^n . Il numero di programmi di lunghezza strettamente inferiore a n è $\sum_{k=0}^{n-1} 2^k = 2^n - 1$. Poiché un programma di lunghezza k può produrre al

massimo una stringa, il numero di stringhe di lunghezza n con $C(x) < n$ è al più $2^n - 1$. Quindi:

$$\#\{x \in \{0,1\}^n : C(x) < n - k\} \leq 2^{n-k}$$

e la probabilità che una stringa uniformemente casuale di lunghezza n abbia complessità inferiore a $n - k$ è al più 2^{-k} . Per $k = 10$, questa probabilità è inferiore a $1/1024$: meno dell'1 per mille delle stringhe di qualunque lunghezza sono comprimibili di più di 10 bit. Quasi tutte le stringhe sono essenzialmente incompressibili — non hanno struttura descrivibile.

Questo risultato ha una conseguenza filosofica precisa: la struttura è l'eccezione, non la regola. Il mondo fisico osservato — con le sue leggi, le sue regolarità, la sua struttura a larga scala — è un sottoinsieme straordinariamente atipico dello spazio delle configurazioni possibili. L'intelligenza, biologica o artificiale, è essenzialmente un sistema di rilevamento e sfruttamento delle regolarità strutturali del mondo: opera nel sottoinsieme di bassa complessità dell'universo delle possibilità.

A.3 L'incompletezza come incompressibilità: il teorema di Chaitin

Il collegamento tra la complessità algoritmica e i teoremi di incompletezza è stabilito da un risultato di Gregory Chaitin che ha la stessa struttura della dimostrazione gödeliana ma in un linguaggio radicalmente diverso. Definiamo il **numero di Chaitin** Ω come:

$$\Omega = \sum_{p : U(p) \text{ termina}} 2^{-|p|}$$

dove la somma è su tutti i programmi p che, eseguiti su U , terminano. Ω è la probabilità che U termini su un programma scelto uniformemente a caso da tutti i programmi finiti — è la *probabilità di fermata* di U . Ω è un

numero reale nell'intervallo $(0,1)$ ben definito, ma ha proprietà straordinarie.

Prima proprietà: Ω è **calcolabile da sinistra** (*left-computable*, o *lower semi-computable*): esiste un algoritmo che produce approssimazioni sempre migliori a Ω per eccesso. Basta enumerare tutte le coppie (programma, tempo di esecuzione), eseguire ogni programma per il numero di passi indicato, e sommare $2^{-|p|}$ ogni volta che un programma termina. Questa somma parziale cresce monotonamente verso Ω .

Seconda proprietà: Ω è **algoritmicamente casuale** nel senso di Martin-Löf: la sequenza binaria dei bit di Ω è incompressibile, ossia $C(\Omega_1\Omega_2...\Omega_n) \geq n - O(1)$ per ogni n , dove $\Omega_1\Omega_2...$ sono le cifre binarie di Ω . Non esiste alcun algoritmo in grado di produrre i bit di Ω più efficientemente dell'enumerazione diretta.

Terza proprietà — il teorema di incompletezza di Chaitin: per ogni sistema formale S coerente e con assiomi ricorsivamente enumerabili, esiste una costante L_S tale che S non può dimostrare alcun enunciato della forma:

$$C(x) > L_S$$

ossia S non può dimostrare che *nessuna* stringa specifica ha complessità superiore a L_S . Informalmente: un sistema formale di complessità data non può dimostrare che un oggetto è significativamente più complesso del sistema stesso.

La dimostrazione sfrutta la stessa struttura diagonale di Gödel. Supponiamo che S possa dimostrare $C(x) > L$ per qualche L grande. Costruiamo un programma che enumera tutte le dimostrazioni di S cercando una dimostrazione di un enunciato $C(x) > L$ per qualche stringa x , e output quella stringa x quando la trova. Questo programma

ha lunghezza circa $\log_2 |S| + O(1)$ — la dimensione della descrizione di S più una costante. Se L è molto maggiore di $\log_2 |S|$, allora il programma produce una stringa di complessità $> L$ ma ha lunghezza $\ll L$, contraddicendo la definizione di complessità di Kolmogorov. Dunque L non può essere troppo grande: è limitato dalla complessità del sistema S stesso.

Il teorema di incompletezza di Chaitin ha una lettura intuitiva di grande eleganza: un sistema formale non può certificare la complessità di oggetti più complessi di se stesso. La logica non può "vedere oltre se stessa" precisamente perché il concetto di complessità codifica tutta l'informazione computazionale, e un sistema di complessità data non può certificare la presenza di complessità superiore. Questo è il correlato algoritmico dell'incompletezza gödeliana, formulato nel linguaggio dell'informazione invece che in quello della provabilità.

A.4 La complessità condizionale e la mutua informazione algoritmica

La teoria della complessità algoritmica si estende a nozioni condizionali e relazionali che rispecchiano le nozioni corrispondenti della teoria dell'informazione di Shannon. La **complessità condizionale** $C(x|y)$ è la lunghezza del programma più corto che, dato y come input ausiliario, produce x :

$$C(x|y) = \min\{|p| : U(p, y) = x\}$$

e la **mutua informazione algoritmica** tra x e y è:

$$I(x : y) = C(x) - C(x|y) = C(y) - C(y|x) + O(\log(C(x) + C(y)))$$

dove l'uguaglianza vale a meno di termini logaritmici, a differenza della mutua informazione di Shannon che vale con uguaglianza esatta. La

simmetria della mutua informazione algoritmica — il fatto che la quantità di informazione di x in y sia uguale alla quantità di informazione di y in x — è un teorema non banale che richiede il teorema di invarianza per la sua dimostrazione.

Questa struttura ha una rilevanza diretta per la teoria dell'apprendimento: un sistema che apprende da esempi sta essenzialmente comprimendo l'informazione degli esempi in un modello più corto. La qualità del modello appreso può essere misurata dalla lunghezza della sua descrizione più la lunghezza dei dati codificati usando il modello — il **MDL (Minimum Description Length) principle** di Rissanen (1978), che formalizza l'intuizione occamiana della parsimonia come principio di inferenza.

A.5 L'induzione di Solomonoff: la predizione bayesiana universale

Il contributo di Ray Solomonoff alla teoria della complessità algoritmica è il meno noto dei tre fondatori del campo ma è forse il più rilevante per la teoria dell'apprendimento. Nel suo articolo del 1964 *A Formal Theory of Inductive Inference*, Solomonoff formulò quello che è oggi chiamato **prior universale di Solomonoff** e il **predittore universale di Solomonoff**.

Il punto di partenza è la domanda: qual è la distribuzione di probabilità a priori "corretta" su tutte le possibili sequenze future, data una sequenza osservata? L'intuizione bayesiana dice che occorre un prior sulle ipotesi e che il prior dovrebbe riflettere qualcosa di fondamentale sulla struttura del mondo. Solomonoff propose che il prior corretto sia determinato dalla complessità algoritmica.

Definiamo la **misura universale di Solomonoff** $M(x)$ come:

$$M(x) = \sum_{p: U(p) \text{ inizia con } x} 2^{-|p|}$$

ossia la probabilità che U , eseguito su un programma scelto uniformemente a caso bit per bit, produca una stringa che inizia con x . $M(x)$ assegna probabilità alta alle stringhe che hanno molte descrizioni corte — alle stringhe di bassa complessità algoritmica — e probabilità bassa alle stringhe di alta complessità.

Il **teorema di convergenza di Solomonoff** afferma: sia P una qualsiasi distribuzione di probabilità computabile su sequenze infinite. Allora esiste una costante c_P tale che per ogni sequenza $x_1x_2\dots x_n\dots$:

$$\sum_{n=1}^{\infty} (P(x_1\dots x_n) - M(x_1\dots x_n))^2 \leq c_P \cdot \ln 2$$

dove la somma è presa lungo le sequenze generate da P con probabilità uno. In parole: il predittore di Solomonoff M converge al predittore ottimale P per qualunque distribuzione computabile P , e la somma quadratica degli errori di previsione è limitata da una costante che dipende soltanto dalla complessità di P , non dalla lunghezza della sequenza.

Questo è il risultato più forte possibile nella teoria della predizione: il predittore universale di Solomonoff è asintoticamente ottimale rispetto a qualunque distribuzione computabile sul mondo. Non può fare peggio di qualunque altro predittore in modo sistematico, nel senso che la somma quadratica delle differenze nelle previsioni è finita. Il prior universale non assume nulla sulla struttura del mondo — non assume linearità, stazionarietà, Markovianità, o qualunque altra proprietà — eppure converge al predittore ottimale per qualunque mondo computabile.

Il prezzo di questa potenza è la non-computabilità: $M(x)$ non è calcolabile, perché richiederebbe di risolvere il problema della fermata. Il predittore di Solomonoff è un ideale teorico — la migliore risposta possibile alla domanda "come si dovrebbe predire razionalmente" — ma non è implementabile. I sistemi neurali reali approssimano questo ideale

nella misura in cui i loro parametri codificano una distribuzione che si avvicina al prior universale sui dati di addestramento. Il gradient descent è un mezzo pratico per navigare verso questa approssimazione.

A.6 La connessione con il criterio MDL e la regolarizzazione

Il principio MDL di Rissanen fornisce la connessione operativa tra la complessità algoritmica e l'apprendimento statistico pratico. Il principio afferma: dato un insieme di dati D e una classe di modelli $\mathcal{M}$, il modello ottimale è quello che minimizza la lunghezza totale della descrizione:

$$L(M) + L(D | M)$$

dove $L(M)$ è la lunghezza della descrizione del modello M e $L(D | M)$ è la lunghezza della descrizione dei dati D dato il modello M — la lunghezza del codice per i dati sotto il modello. Questa somma è la lunghezza della descrizione più corta dell'intero sistema modello-dati.

Il criterio MDL ha una corrispondenza formale con la regolarizzazione in statistica. La *regolarizzazione di Tikhonov* aggiunge un termine di penalità alla funzione di perdita:

$$\mathcal{L}_{\text{reg}}(\theta) = \mathcal{L}(\theta) + \lambda \cdot \|\theta\|^2$$

dove λ è il coefficiente di regolarizzazione e $\|\theta\|^2$ penalizza la norma dei parametri. Nella cornice MDL, il termine di penalità corrisponde a $L(M)$ — la lunghezza della descrizione del modello — e il termine di perdita corrisponde a $L(D | M)$ — la lunghezza dei dati dato il modello. Scegliere λ è equivalente a scegliere una distribuzione a priori sulla complessità dei modelli.

Nella prospettiva della complessità algoritmica, la regolarizzazione è il meccanismo attraverso cui i sistemi apprendenti privilegiano le ipotesi di bassa complessità, approssimando il prior universale di Solomonoff con

una famiglia parametrizzata di prior più semplici. I modelli neurali moderni non usano esplicitamente la regolarizzazione di Tikhonov, ma la SGD con minibatch esercita una forma di regolarizzazione implicita — già evocata nella Parte Terza — che seleziona soluzioni di norma minima tra i parametri che minimizzano la perdita sull'insieme di addestramento. Questa regolarizzazione implicita ha la struttura formale di una preferenza per ipotesi di bassa complessità, in accordo con il principio MDL.

A.7 Le leggi di scala come struttura della complessità

Un risultato empirico di grande rilevanza teorica per la connessione tra complessità algoritmica e apprendimento neurale è quello delle **leggi di scala** (*scaling laws*) documentate da Kaplan, McCandlish, Henighan e colleghi di OpenAI nel 2020. Lo studio ha mostrato che la perdita di un modello linguistico sull'insieme di test segue leggi di potenza rispetto a tre parametri: il numero di parametri N del modello, la dimensione del corpus di addestramento D in token, e il budget computazionale C in FLOP:

$$L(N) \approx (N_c / N)^{\alpha_N} \text{ con } \alpha_N \approx 0.076 \quad L(D) \approx (D_c / D)^{\alpha_D} \\ \text{con } \alpha_D \approx 0.095 \quad L(C) \approx (C_c / C)^{\alpha_C} \text{ con } \alpha_C \approx 0.050$$

dove N_c , D_c , C_c sono costanti di normalizzazione e i parametri α definiscono la pendenza della legge di potenza nel piano log-log. Queste relazioni valgono su più di sei ordini di grandezza in ciascuna delle variabili.

Le leggi di potenza nella complessità computazionale e nella teoria dell'informazione non sono accidentali: emergono tipicamente quando il sistema sta estraendo struttura da una distribuzione che ha a sua volta struttura frattale o auto-simile. Una distribuzione che soddisfa leggi di

potenza ha la proprietà che la struttura ai piccoli scale rispecchia la struttura ai grandi scale — è una distribuzione senza scala caratteristica. La distribuzione delle frequenze lessicali nei testi naturali segue la legge di Zipf — frequenza $\propto 1/\text{rango}$ — che è una legge di potenza. La distribuzione delle lunghezze delle frasi, delle dipendenze sintattiche, delle catene di coreferenza: tutte seguono approssimativamente leggi di potenza. Un modello addestrato su questa distribuzione deve apprendere strutture a tutte le scale, e la legge di scala della perdita riflette la struttura informazionale gerarchica del linguaggio naturale.

Nella prospettiva della complessità algoritmica, le leggi di scala suggeriscono che la struttura del linguaggio naturale ha una compressibilità che cresce monotonamente con le risorse — ogni ordine di grandezza in più di parametri o di dati estrae un'ulteriore frazione della struttura regolare del linguaggio. Non c'è una soglia al di là della quale tutta la struttura è stata estratta: il linguaggio umano sembra avere struttura regolare a tutte le scale osservate. Questo è coerente con l'ipotesi che la complessità algoritmica del linguaggio umano sia molto alta — il linguaggio è uno degli oggetti più strutturati e al tempo stesso più ricchi che esistano nella cultura umana.

A.8 L'incompletezza come limite dell'apprendimento

Il teorema di incompletezza di Chaitin, letto in chiave di teoria dell'apprendimento, produce un risultato negativo di notevole portata: nessun sistema apprendente può apprendere a riconoscere in modo affidabile la complessità algoritmica. Più precisamente, nessun algoritmo può certificare, per ogni stringa x , se $C(x) > k$ per una soglia k arbitrariamente grande.

Questo ha conseguenze dirette per i sistemi neurali. Un modello linguistico addestrato su testi umani apprende a modellare la distribuzione dei testi umani — che ha struttura regolare e bassa complessità algoritmica relativa alla dimensione dei testi stessi. Ma il modello non può apprendere a riconoscere la complessità algoritmica assoluta delle stringhe, perché questa proprietà è non-calcolabile. Il modello può essere un ottimo predittore della continuazione di testi umani senza essere capace di distinguere tra una stringa genuinamente casuale e una stringa pseudocasuale generata da un algoritmo di complessità moderata.

Questo è il limite fondamentale dei sistemi neurali dal punto di vista della complessità algoritmica: approssimano il prior universale di Solomonoff sulla distribuzione di addestramento, ma non possono superare i limiti computazionali imposti dalla non-computabilità di Ω . L'incompletezza di Gödel-Chaitin non è elusa dai sistemi neurali: è aggirata in modo pragmatico, confinando l'attenzione alla sottoclasse computabile e strutturata del mondo, che è esattamente la sottoclasse che i dati di addestramento campionano.

Espansione — Parte B

Apprendibilità, generalizzazione e la preistoria filosofica del collasso simbolico

B.1 La teoria PAC: cosa è apprendibile in linea di principio

La domanda "cosa possono apprendere i sistemi neurali?" non ha ricevuto, nella monografia originale, la risposta formale che merita. Il quadro che fornisce questa risposta è la **Probably Approximately**

Correct (PAC) learning theory, formulata da Leslie Valiant nel fondamentale articolo del 1984 *A Theory of the Learnable* — un lavoro che ha fondato la teoria computazionale dell'apprendimento come disciplina autonoma e per il quale Valiant ha ricevuto il premio Turing nel 2010.

Il punto di partenza è una precisazione di cosa si intende per "apprendere". L'apprendimento è una relazione tra tre componenti: una classe di concetti $\mathbb{C}$ (gli oggetti che si vogliono imparare a riconoscere), un algoritmo di apprendimento A , e una distribuzione di probabilità D sullo spazio degli esempi X . Un concetto $c \in \mathbb{C}$ è una funzione $c: X \rightarrow \{0,1\}$ che classifica ogni esempio come positivo o negativo. Il classificatore ideale è c^* — il concetto target.

L'algoritmo A riceve un insieme di addestramento $S = \{(x_1, c^*(x_1)), \dots, (x, c^*(x))\}$ dove gli x_i sono estratti i.i.d. dalla distribuzione D . A produce un'ipotesi $h: X \rightarrow \{0,1\}$ che spera di approssimare c^* . La qualità dell'ipotesi è misurata dall'**errore di generalizzazione**:

$$\text{err}_D(h) = P_{x \sim D} [h(x) \neq c^*(x)]$$

ossia la probabilità che h classifichi incorrettamente un nuovo esempio estratto da D .

La definizione PAC è la seguente: una classe di concetti $\mathbb{C}$ è **PAC-apprendibile** se esiste un algoritmo A tale che per ogni $\epsilon > 0$, $\delta > 0$, ogni distribuzione D su X , e ogni concetto target $c^* \in \mathbb{C}$, quando A riceve $m \geq m(\epsilon, \delta)$ esempi estratti da D e etichettati secondo c^* , produce con probabilità almeno $1 - \delta$ un'ipotesi h con $\text{err}_D(h) \leq \epsilon$. Il termine "probably" corrisponde alla confidenza $1 - \delta$, il termine "approximately correct" all'errore ϵ .

La **sample complexity** $m(\epsilon, \delta)$ — il numero minimo di esempi necessari per garantire apprendimento PAC — è il parametro fondamentale della

teoria: misura quanto dati un algoritmo di apprendimento ha bisogno per imparare un concetto con accuratezza e confidenza specificate.

B.2 La dimensione di Vapnik-Chervonenkis

Il carattere che determina la PAC-apprendibilità di una classe di concetti è la **dimensione di Vapnik-Chervonenkis (VC)**, introdotta da Vladimir Vapnik e Alexei Chervonenkis nel 1971. Questo numero cattura la "capacità espressiva" della classe di ipotesi in un senso geometrico preciso.

Definizione. Sia $\mathcal{H}$ una classe di ipotesi binarie su X . Un insieme finito $S = \{x_1, \dots, x\} \subset X$ è **shattered** (spezzato) da $\mathcal{H}$ se per ogni possibile assegnazione di etichette binarie $(b_1, \dots, b) \in \{0,1\}^m$ esiste un'ipotesi $h \in \mathcal{H}$ tale che $h(x_i) = b_i$ per ogni i . In altri termini, $\mathcal{H}$ può realizzare qualunque classificazione binaria degli elementi di S . La dimensione VC di $\mathcal{H}$ è:

$$\text{VCdim}(\mathcal{H}) = \max\{|S| : S \text{ è spezzato da } \mathcal{H}\}$$

Se $\mathcal{H}$ può spezzare insiemi di dimensione arbitrariamente grande, la dimensione VC è infinita.

Un esempio fondamentale: la classe degli iperpiani lineari in $\mathbb{R}^d$ — classificatori che assegnano label +1 ai punti da un lato dell'iperpiano e -1 all'altro — ha dimensione VC esattamente $d+1$. La dimostrazione della direzione $\text{VCdim} \geq d+1$ è costruttiva: si mostra che i $d+1$ vettori della base canonica più il vettore zero formano un insieme spezzato dagli iperpiani. La direzione $\text{VCdim} \leq d+1$ usa il lemma di Radon: ogni insieme di $d+2$ punti in $\mathbb{R}^d$ può essere partizionato in due sottoinsiemi convessamente separabili in modo che la partizione non è realizzabile da tutti gli iperpiani.

B.3 Il teorema fondamentale dell'apprendimento statistico

Il teorema centrale della teoria PAC, il **Teorema Fondamentale dell'Apprendimento Statistico**, stabilisce una corrispondenza perfetta tra dimensione VC finita e PAC-apprendibilità:

Una classe di ipotesi binarie $\mathcal{H}$ è PAC-apprendibile se e solo se la sua dimensione VC è finita.

Per la direzione sufficiente (VC finita implica PAC-apprendibilità), il bound di generalizzazione standard — derivato attraverso il lemma di Sauer e la tecnica dei ghost samples di Vapnik e Chervonenkis — afferma che per ogni distribuzione D , ogni concetto target c^* , e ogni ipotesi $h \in \mathcal{H}$ con errore empirico nullo sull'insieme di addestramento S di dimensione m :

$$P[\text{err}_D(h) > \epsilon] \leq 4 \cdot (2m)^{\text{VCdim}(\mathcal{H})} \cdot e^{-m\epsilon^2/8}$$

Imponendo che questa probabilità sia al più δ e risolvendo per m si ottiene la sample complexity:

$$m(\epsilon, \delta) = O((\text{VCdim}(\mathcal{H})/\epsilon^2) \cdot \log(1/\epsilon) + (1/\epsilon^2) \cdot \log(1/\delta))$$

che è polinomiale in $1/\epsilon$ e logaritmica in $1/\delta$: l'apprendimento richiede un numero di esempi polinomiale nei parametri di accuratezza e confidenza.

Per la direzione necessaria (VC infinita implica non-apprendibilità), la dimostrazione usa un argomento combinatorio: se $\mathcal{H}$ può spezzare insiemi di qualunque dimensione, allora per qualunque algoritmo A e qualunque m , esiste una distribuzione D tale che con probabilità costante A fallisce nell'apprendere il concetto target con accuratezza $1/2$.

B.4 Oltre la PAC: i bound di Rademacher e i risultati di generalizzazione per reti neurali

Il framework PAC classico si applica a classificatori binari deterministici con funzione di perdita 0-1. Per le reti neurali profonde con funzioni di perdita continue e architetture molto più complesse, la teoria della generalizzazione richiede strumenti più sofisticati. Il più potente degli strumenti moderni è la **complessità di Rademacher**, che misura la capacità di una classe di ipotesi di adattarsi al rumore puro.

Definizione. Data una classe di funzioni $\mathcal{F}$ su $X \times Y$ e un campione $S = \{(x_1, y_1), \dots, (x, y)\}$, la complessità di Rademacher empirica di $\mathcal{F}$ su S è:

$$R_S(\mathcal{F}) = E_{\sigma}[\sup_{f \in \mathcal{F}} (1/m) \sum_{i=1}^m \sigma_i f(x_i, y_i)]$$

dove $\sigma = (\sigma_1, \dots, \sigma_m)$ sono variabili i.i.d. di Rademacher — variabili aleatorie uniformi su $\{-1, +1\}$ — e l'aspettazione è rispetto a esse. La complessità di Rademacher misura quanto $\mathcal{F}$ riesce in media a correlare con rumore puro: se $\mathcal{F}$ è molto espressiva, può adattarsi anche al rumore, producendo una complessità alta.

Il **bound di generalizzazione di Rademacher** afferma che con probabilità almeno $1 - \delta$ rispetto al campionamento di S , per ogni $f \in \mathcal{F}$:

$$\text{err}_D(f) \leq \text{err}_S(f) + 2 R_m(\mathcal{F}) + \sqrt{(\log(1/\delta) / (2m))}$$

dove $\text{err}_S(f)$ è l'errore empirico sul campione S e $R_m(\mathcal{F}) = E_S[R_S(\mathcal{F})]$ è la complessità di Rademacher attesa. Il bound è più raffinato del bound VC: dipende dalla distribuzione dei dati attraverso il campione osservato, non solo dalla dimensione della classe di ipotesi.

Per le reti neurali, la complessità di Rademacher dipende dalla norma spettrale delle matrici di pesi — il prodotto dei raggi spettrali attraverso

gli strati — e decresce con questa norma: reti con pesi piccoli generalizzano meglio. Questo è il fondamento teorico della connessione tra la norma dei parametri e la generalizzazione, e spiega perché la regolarizzazione implicita della SGD — che seleziona soluzioni di norma minima — favorisce la generalizzazione.

B.5 Il fenomeno della doppia discesa

Uno dei risultati teorici più sorprendenti degli ultimi anni, che ha richiesto una revisione della comprensione classica dell'overfitting, è il **fenomeno della doppia discesa**, documentato empiricamente da Belkin, Hsu, Ma e Mandal (2019) e analizzato teoricamente da una serie di lavori successivi.

La saggezza convenzionale sull'apprendimento statistico afferma che la relazione tra la complessità del modello e l'errore di test ha la forma di una U — la curva bias-varianza: per modelli poco complessi, l'errore è alto per eccesso di bias (underfitting); aumentando la complessità, l'errore diminuisce fino a un ottimo; per modelli troppo complessi, l'errore risale per eccesso di varianza (overfitting). Questo profilo a U era considerato una legge fondamentale dell'apprendimento.

I dati mostrano invece un profilo più articolato. All'aumentare della capacità del modello — misurata dal numero di parametri N — si osservano due regimi distinti separati da un regime di interpolazione critico:

Nel **regime di sottoparametrizzazione** ($N \ll N_{\text{data}}$), si osserva la classica curva bias-varianza: l'errore di test diminuisce con N fino a un ottimo, poi risale.

Nel **regime di interpolazione critica** ($N \approx N_{\text{data}}$), il modello ha appena abbastanza capacità da interpolare tutti i dati di addestramento

(raggiungere perdita zero). Questo è il punto di picco dell'errore di test nel profilo classico.

Nel **regime di sovrapparametrizzazione** ($N \gg N_{\text{data}}$), l'errore di test *riprende a diminuire* all'aumentare di N , scendendo a livelli inferiori al minimo della curva bias-varianza classica. I modelli enormemente sovrapparametrizzati — come le reti neurali profonde moderne — generalizzano meglio, non peggio, dei modelli di dimensione ottima secondo la teoria classica.

La spiegazione teorica della doppia discesa è connessa alla struttura dello spazio delle soluzioni nella fase di sovrapparametrizzazione. Quando $N \gg N_{\text{data}}$, l'insieme dei parametri che interpolano i dati di addestramento è un manifold di dimensione alta — uno spazio di soluzioni. La SGD, con la sua regolarizzazione implicita, seleziona la soluzione di norma minima in questo manifold. Per N molto grande, questa soluzione di norma minima ha la proprietà che le sue predizioni su nuovi esempi sono vicine alla media delle previsioni di tutte le soluzioni che interpolano i dati — un effetto di *implicit ensemble averaging* che riduce la varianza.

Formalmente, in un modello lineare sovradeterminato — $N > m$ — il regressore di minima norma è lo *pseudoinverso*:

$$\theta^* = X^T(XX^T)^{-1} y$$

dove $X \in \mathbb{R}^{m \times N}$ è la matrice degli esempi e $y \in \mathbb{R}^m$ è il vettore dei label. Il bias di questo stimatore è zero (interpola perfettamente i dati), e la varianza decresce con N come $O(\sigma^2 d/m)$ dove σ^2 è la varianza del rumore e d è la dimensione del segnale utile. Dunque per $N \rightarrow \infty$ la varianza va a zero: il modello di dimensione infinita ha varianza nulla nel limite.

Il fenomeno della doppia discesa ha un'importante implicazione filosofica: l'ottimalità della semplicità — il rasoio di Occam — non è universale. Nei regimi di sovrerparametrizzazione, modelli più complessi generalizzano meglio. Il principio di parsimonia è una buona guida nel regime di sottoparametrizzazione, ma non nel regime in cui operano i grandi modelli linguistici. Questo non confuta il principio MDL di Rissanen — che è formulato in termini di lunghezza descrittiva, non di numero di parametri — ma mostra che la relazione tra complessità del modello e generalizzazione è più sottile di quanto la teoria classica avesse previsto.

B.6 L'apprendimento in-context come inferenza bayesiana implicita

Un fenomeno recente di grande interesse teorico per la tesi della monografia è l'**in-context learning** (*ICL*) dei modelli linguistici transformer: la capacità di apprendere un nuovo compito semplicemente osservando alcuni esempi del compito nel prompt, senza aggiornare i pesi del modello. Questa capacità, documentata sistematicamente per i modelli GPT-3 e successivi, è teoricamente sorprendente perché realizza un apprendimento senza gradient descent.

Il lavoro di Xie, Raghunathan, Liang e Ma (2021) *An Explanation of In-Context Learning as Implicit Bayesian Inference* fornisce un'interpretazione formale: l'ICL è equivalente all'inferenza bayesiana implicita sul concetto latente che genera gli esempi osservati nel contesto.

Il modello teorico è il seguente. Si suppone che esista un concetto latente c^* che determina la mappatura input-output negli esempi del prompt: $(x_1, y_1), \dots, (x, y)$ sono i.i.d. da $P(\cdot | c^*)$. Il modello, addestrato su documenti che contengono pattern simili, ha internamente un prior

implicito $P(c)$ sui concetti. Osservando gli esempi nel prompt, il modello aggiorna implicitamente questo prior secondo la regola di Bayes:

$$P(c \mid x_{1:k}, y_{1:k}) \propto P(y_{1:k} \mid x_{1:k}, c) \cdot P(c)$$

e predice il label del nuovo esempio x_{+1} come:

$$P(y_{+1} \mid x_{+1}, x_{1:k}, y_{1:k}) = \int P(y_{+1} \mid x_{+1}, c) P(c \mid x_{1:k}, y_{1:k}) dc$$

Xie e colleghi dimostrano che un modello preaddestrato sulla distribuzione $P(\text{documento})$ — dove i documenti sono sequenze generate da concetti latenti con distribuzione $P(c)$ — apprende implicitamente il prior $P(c)$ e può eseguire l'aggiornamento bayesiano quando vede esempi nel contesto. L'ICL non è un algoritmo separato dall'addestramento: è la manifestazione, al momento dell'inferenza, della struttura bayesiana acquisita durante l'addestramento.

Questo risultato rafforza la tesi centrale della monografia in modo notevole: i modelli linguistici non solo approssimano la distribuzione dei testi umani, ma apprendono implicitamente la struttura inferenziale bayesiana che gli esseri umani usano per generalizzare da pochi esempi. L'inferenza bayesiana non è un algoritmo separato che viene eseguito sopra il modello: emerge dalla struttura statistica del corpus e si manifesta come proprietà dell'architettura transformer addestrata.

Il lavoro di Akyürek, Schuurmans, Andreas, Ma e Zhou (2022) *What Learning Algorithm is In-Context Learning? Investigations with Linear Models* ha mostrato, in un setting semplificato di regressione lineare, che i transformer addestrati su compiti di regressione realizzano implicitamente gradient descent: le attivazioni interne del modello durante l'ICL seguono la stessa traiettoria che seguirebbe un algoritmo di gradient descent eseguito sugli esempi del prompt. Questo suggerisce che i transformer abbiano imparato a *implementare algoritmi di apprendimento*

nei propri parametri — una forma di meta-apprendimento (*learning to learn*) che emerge dall'addestramento su distribuzioni di compiti diversi.

B.7 Il Circolo di Vienna e la preistoria filosofica del collasso simbolico

Il programma hilbertiano nella matematica e il programma dell'AI simbolica di Newell e Simon non erano eventi isolati: si inserivano in un movimento intellettuale più ampio che aveva il suo cuore filosofico nel **Circolo di Vienna** (*Wiener Kreis*) degli anni Venti e Trenta. Comprendere il collasso di quel movimento è necessario per capire le ragioni profonde del collasso dell'AI simbolica, che ne condivideva le assunzioni fondamentali.

Il Circolo di Vienna — fondato intorno a Moritz Schlick, con figure centrali come Rudolf Carnap, Otto Neurath, Hans Hahn, Friedrich Waismann, e visitatori regolari come Kurt Gödel e Ludwig Wittgenstein nella fase iniziale — era animato da un programma di **riforma radicale della filosofia** ispirata alla precisione della logica e della matematica. Il manifesto programmatico *Wissenschaftliche Weltauffassung: Der Wiener Kreis* del 1929 proclamava: la filosofia non è una disciplina autonoma con metodi propri, ma l'analisi logica del linguaggio della scienza. I problemi filosofici tradizionali — l'esistenza di Dio, la natura della coscienza, il fondamento della morale — non sono problemi genuini: sono pseudoproblemi generati da un uso impreciso del linguaggio. L'analisi logica li dissolve mostrando che i loro enunciati non hanno contenuto empiricamente verificabile.

Il principio fondamentale era il **criterio empiristico di significato** nella formulazione di Carnap: un enunciato è cognitivamente significativo se e solo se è verificabile in linea di principio attraverso l'esperienza — o è

analitico (vero per definizione, come le verità logiche e matematiche). Tutto il resto è emotivamente espressivo ma cognitivamente privo di senso.

Carnap portò questo programma alle sue conseguenze più rigorose nel volume *Der logische Aufbau der Welt* del 1928: tentò di costruire, partendo dalle esperienze elementari soggettive e usando la logica simbolica dei *Principia Mathematica*, l'intero edificio della conoscenza empirica attraverso definizioni esplicite. Il mondo doveva essere *costituito logicamente* a partire dagli elementi empirici attraverso connettivi logici, quantificatori e relazioni. Era la versione epistemologica del fondazionalismo logico hilbertiano: così come Hilbert voleva derivare tutta la matematica da assiomi elementari, Carnap voleva derivare tutta la conoscenza empirica da esperienze elementari attraverso connessioni logiche esplicite.

Il programma carnapiano subì lo stesso destino di quello hilbertiano, attraverso due colpi distinti. Il primo — interno — fu il riconoscimento, avanzato da Quine nel saggio *Two Dogmas of Empiricism* del 1951, che la distinzione tra enunciati analitici e sintetici — la distinzione su cui si fondava l'intero programma del Circolo — non è tracciabile in modo netto. Non esiste una lista fissa di enunciati analitici: la distinzione dipende dall'intero sistema di credenze, che è soggetto a revisione olistica. La conoscenza non si costruisce pezzo per pezzo a partire da fondamenti certi: è una rete (*web of belief*) in cui ogni nodo è in linea di principio revisionabile alla luce di evidenze sufficientemente forti.

Il secondo colpo — esterno — fu il problema del *grue* di Nelson Goodman (1955): anche dato un insieme finito di osservazioni, esistono sempre infiniti predicati compatibili con quelle osservazioni che fanno previsioni diverse sulle osservazioni future. La logica non determina quale proiezione fare: c'è sempre un residuo di scelta che nessuna analisi

logica può eliminare. Questo è il correlato epistemologico del frame problem nell'AI: così come la rappresentazione simbolica non può specificare a priori tutto ciò che rimane invariato dopo un'azione, la logica non può specificare a priori quale proiezione fare da un insieme finito di casi al caso generale.

Il collasso del programma del Circolo di Vienna e il collasso dell'AI simbolica non sono accidenti storici paralleli: sono manifestazioni della stessa difficoltà strutturale. Entrambi i programmi presupponevano che la conoscenza potesse essere completamente articolata in un sistema formale esplicito — un sistema di regole, di assiomi, di definizioni — e che l'intelligenza fosse essenzialmente l'applicazione sistematica di questo sistema. La struttura aperta della conoscenza pratica — la texture aperta nel senso di Waismann — ha mostrato in entrambi i casi che questo presupposto è falso.

B.8 Il problema della stanza cinese e la questione semantica aperta

Nessuna trattazione onesta delle implicazioni filosofiche dell'AI può ignorare l'argomento della **stanza cinese** di John Searle, presentato nel saggio del 1980 *Minds, Brains, and Programs* in *Behavioral and Brain Sciences*. L'argomento è stato spesso liquidato in modo troppo rapido sia dai suoi sostenitori sia dai suoi critici, e merita di essere preso sul serio nella sua forma più forte.

Searle immagina un parlante di inglese chiuso in una stanza con un grande manuale di regole per manipolare simboli cinesi: il manuale specifica, per ogni sequenza di simboli cinesi in input, quale sequenza di simboli cinesi produrre in output. Il sistema — l'uomo più il manuale — risponde alle domande in cinese in modo indistinguibile da un parlante

nativo. Eppure, argomenta Searle, l'uomo nella stanza non capisce il cinese: manipola simboli senza avere alcuna comprensione del loro significato. La sintassi — la manipolazione formale dei simboli — non è sufficiente per la semantica — il significato genuino.

L'argomento è diretto contro il **funzionalismo forte**: la tesi che la cognizione sia definita dalla struttura funzionale del processo (l'input-output mapping) indipendentemente dal substrato fisico che la implementa. Searle sostiene che la struttura funzionale è necessaria ma non sufficiente per la comprensione genuina: occorre qualcosa di più, che Searle chiama i "poteri causali" del cervello biologico — una proprietà che Searle non specifica con precisione, il che rende la sua posizione vulnerabile alla critica di essere misteriosa.

Le risposte standard all'argomento della stanza cinese sono tre. La *risposta del sistema* (Dennett, Hofstadter) afferma che la comprensione non appartiene all'uomo nella stanza ma al sistema nella sua interezza — l'uomo più il manuale. Searle la respinge osservando che anche se l'uomo memorizzasse l'intero manuale, continuando ad eseguire meccanicamente le operazioni, non capirebbe il cinese.

La *risposta del robot* afferma che un sistema come quello della stanza cinese, ma incorporato in un robot con sensori e attuatori connessi al mondo, svilupperebbe comprensione semantica attraverso l'ancoraggio percettivo dei simboli nel mondo — il *symbol grounding* di Harnad. Questa risposta è più seria: suggerisce che la semantica richiede una connessione causale tra i simboli e gli stati del mondo che essi rappresentano, connessione che manca nel sistema della stanza cinese.

La *risposta del cervello simulato* afferma che se il sistema della stanza cinese simulasse fedelmente ogni neurone del cervello di un parlante di cinese, avrebbe comprensione. Searle respinge anche questa risposta,

argomentando che la simulazione non è la cosa simulata: simulare l'incendio di un edificio non brucia nulla.

Il punto filosofico che rimane aperto dopo questo dibattito è il seguente: l'argomento della stanza cinese mostra con forza che la manipolazione sintattica formale non è sufficiente per la comprensione semantica. Ma non stabilisce dove si trova esattamente il confine: non chiarisce quali proprietà aggiuntive — incarnazione, connessione causale con il mondo, continuità temporale, coscienza — sono necessarie per la comprensione genuina. Questo confine rimane tra i problemi filosofici aperti più profondi della filosofia della mente.

Per la tesi di questa monografia, il punto rilevante è che l'argomento della stanza cinese si applica con forza ai sistemi simbolici classici — la stanza cinese è esattamente un sistema esperto basato su regole — e molto meno chiaramente ai sistemi neurali statistici. I modelli linguistici neurali non manipolano simboli secondo un manuale esplicito di regole: producono distribuzioni di probabilità su sequenze, dove la struttura della distribuzione emerge dall'addestramento su milioni di esempi di comunicazione umana radicata nel mondo. Se la comprensione richiede connessione causale con il mondo, i modelli neurali addestrati su testi prodotti da agenti causalmente connessi al mondo hanno almeno una connessione indiretta attraverso il corpus. Questo non risolve il problema, ma lo riformula in modo sostanzialmente diverso rispetto alla stanza cinese originale.

B.9 La realizzabilità multipla e l'irriducibilità del livello funzionale

Un argomento classico del funzionalismo che acquista nuova forza alla luce della matematica dell'AI è la **realizzabilità multipla** (*multiple*

realizability)), formulata da Hilary Putnam nel 1967. L'argomento è il seguente: il dolore — e più in generale qualunque stato mentale — può essere realizzato da substrati fisici diversi: neuroni nel cervello umano, forse silicio in un sistema artificiale, forse qualunque altro substrato fisico organizzato nel modo funzionale appropriato. Se il dolore fosse identico a uno stato neurale specifico, non potrebbe essere realizzato da substrati diversi. Quindi il dolore non è identico a uno stato neurale: è uno stato funzionale definito dal suo ruolo causale, indipendente dal substrato.

La realizzabilità multipla ha una dimostrazione di fatto nei sistemi AI: le stesse funzioni cognitive — riconoscimento del parlato, traduzione, ragionamento visivo — sono realizzate sia nei cervelli biologici sia nelle reti neurali artificiali, con architetture e substrati radicalmente diversi. Questo suggerisce che esiste un livello di descrizione funzionale che è irriducibile al livello fisico: il livello computazionale di Marr, che specifica cosa il sistema calcola indipendentemente da come lo calcola.

L'irriducibilità del livello funzionale ha un corollario importante per la questione della logica: se le capacità cognitive sono proprietà funzionali irriducibili al substrato, allora la capacità di ragionamento logico è una proprietà funzionale — qualcosa che il sistema fa — non una proprietà del substrato. La domanda "il sistema ragiona logicamente?" è una domanda sul livello funzionale, e la risposta non dipende da se il substrato è neurale o artificiale. Ma la domanda "qual è il meccanismo attraverso cui il sistema realizza la capacità di ragionamento logico?" è una domanda sul livello algoritmico e implementativo, e la risposta — come si è argomentato nell'intera monografia — è che il meccanismo è statistico e distributivo, non logico-deduttivo.

Questa separazione tra il livello funzionale e il livello implementativo dissolve una confusione che ha afflitto molto del dibattito sull'AI: la

confusione tra la capacità di produrre output logicamente coerenti (livello funzionale) e l'implementazione di un motore di inferenza logica (livello implementativo). I sistemi neurali hanno la prima capacità senza la seconda architettura. E poiché la capacità è ciò che importa per la cognizione, mentre l'architettura implementativa è una scelta ingegneristica, i sistemi neurali realizzano — nel senso tecnico di Putnam — la cognizione logica senza essere sistemi logici.

Espansione — Parte C

Co-evoluzione gene-cultura, effetto Baldwin e il radicamento semantico come minimizzazione dell'energia libera

C.1 Il problema della trasmissione culturale delle competenze formali

La Parte Quarta della monografia ha argomentato che il ragionamento logico formale è una tecnologia cognitiva acquisita attraverso pratiche culturali, non un meccanismo nativo selezionato dall'evoluzione. Questa tesi lascia aperta una domanda di notevole profondità: come interagisce la trasmissione culturale delle competenze con l'evoluzione biologica del substrato che le acquisisce? La risposta a questa domanda richiede la teoria della **co-evoluzione gene-cultura**, sviluppata sistematicamente da Cavalli-Sforza e Feldman nel volume *Cultural Transmission and Evolution* del 1981 e da Boyd e Richerson nel volume *Culture and the Evolutionary Process* del 1985, e integrata con il meccanismo dell'**effetto Baldwin**, proposto indipendentemente da James Mark Baldwin, Lloyd Morgan e Henry Fairfield Osborn nel 1896.

La questione cruciale è la seguente: se una competenza cognitiva come il ragionamento logico viene acquisita culturalmente, la selezione naturale non opera su di essa direttamente — opera sui geni, non sulle competenze apprese. Come può dunque la trasmissione culturale di competenze influenzare l'evoluzione biologica del cervello? La risposta standard è che la cultura modifica il **paesaggio adattivo** — la funzione che assegna a ogni genotipo un valore di fitness — e quindi altera indirettamente la direzione della selezione.

L'effetto Baldwin descrive precisamente questo meccanismo. Supponiamo che in una popolazione esista variabilità genetica nella facilità di acquisire una certa competenza culturale — per esempio, la capacità di apprendere sistemi di rappresentazione simbolica. Gli individui con genotipi che facilitano l'acquisizione rapida di questa competenza hanno vantaggi riproduttivi in un ambiente culturale in cui la competenza è utile. La selezione favorisce dunque i genotipi che abbassano il costo di acquisizione della competenza, fino al limite in cui la competenza diventa così facile da acquisire da sembrare innata — anche se non è determinata geneticamente in modo diretto.

C.2 Il modello formale della co-evoluzione gene-cultura

Il modello matematico di base della co-evoluzione gene-cultura opera su due sistemi di trasmissione paralleli con dinamiche distinte. La trasmissione genetica segue la dinamica mendeliana standard: un allele A con frequenza p nella generazione t ha frequenza nella generazione $t+1$ determinata dalla fitness relativa w_A :

$$p_{\{t+1\}} = p_t \cdot w_A / [p_t \cdot w_A + (1 - p_t) \cdot w_a]$$

dove w_a è la fitness dell'allele alternativo a . La trasmissione culturale è più flessibile: un tratto culturale può essere trasmesso verticalmente (dai

genitori ai figli), orizzontalmente (tra coetanei) o obliquamente (da individui di generazioni precedenti non parentali). Cavalli-Sforza e Feldman rappresentano la trasmissione culturale attraverso una matrice di transizione:

$$P(\text{tratto in figlio} \mid \text{tratto in genitore 1, tratto in genitore 2}) = b + (1-b) \cdot f_{\text{bias}}$$

dove b è la probabilità di trasmissione fedele e f_{bias} cattura effetti di conformismo, prestigio, e preferenze individuali. La distinzione fondamentale con la trasmissione genetica è la possibilità di trasmissione orizzontale e di modificazione dell'individuo nel corso della sua vita: i tratti culturali possono cambiare nei vivi, non solo nelle generazioni successive.

L'interazione tra i due sistemi produce la co-evoluzione. Sia X la frequenza di un allele che facilita l'acquisizione di un tratto culturale T , e sia f la frequenza del tratto culturale T nella popolazione. Le equazioni di co-evoluzione hanno la struttura:

$$dX/dt = X(1 - X) \cdot [w(X, f) - \bar{w}] \quad df/dt = g(f, X) - d \cdot f$$

dove $w(X, f)$ è la fitness dell'allele in funzione sia del genotipo sia della frequenza culturale, $\bar{w}$ è la fitness media nella popolazione, $g(f, X)$ è la funzione di acquisizione culturale che dipende dalla frequenza del tratto e dalla facilità genetica di acquisirlo, e d è il tasso di perdita del tratto culturale. Il sistema può avere punti fissi stazionari in cui sia la frequenza genica che quella culturale si stabilizzano, oppure cicli o dinamiche più complesse.

Il modello di Boyd e Richerson aggiunge la **copiatura delle tendenze** (*biased transmission*): gli individui non copiano i tratti culturali in modo casuale, ma preferiscono copiare individui di alto prestigio (*prestige bias*) o

tratti frequenti nella popolazione (*conformist bias*). L'effetto conformistico è matematicamente importante: produce effetti di soglia in cui tratti culturali rari possono essere soppressi anche se adattativi, mentre tratti diffusi sono rinforzati anche se subottimali. Questo spiega la persistenza di norme culturali inefficienti e la difficoltà di diffusione di innovazioni radicali — incluse le rivoluzioni concettuali nella logica e nella matematica.

C.3 L'effetto Baldwin e la genetizzazione delle competenze

L'effetto Baldwin descrive il processo attraverso cui una competenza appresa culturalmente può, nel corso di generazioni, lasciare un'impronta nell'architettura genetica del cervello — non come codifica diretta della competenza, ma come abbassamento del costo di acquisizione. Il meccanismo è indiretto: la selezione favorisce gli individui che apprendono più facilmente la competenza adattiva, e questo seleziona per un cervello i cui vincoli architetturali si adattano meglio all'acquisizione della competenza in questione.

Il modello matematico elementare dell'effetto Baldwin considera una popolazione in cui la fitness di un individuo dipende dalla probabilità di acquisire una competenza culturale C nel corso della vita. Sia α la probabilità di acquisire C in assenza di predisposizione genetica, e sia $\alpha + \delta_G$ la probabilità di acquisire C per individui con genotipo G favorevole. La fitness media degli individui con genotipo G è:

$$w_G = w_{\text{base}} + \Delta w \cdot (\alpha + \delta_G)$$

dove Δw è il vantaggio fitness derivante dall'acquisire la competenza. La selezione favorisce G con intensità:

$$s_G = w_G - \bar{w} \approx \Delta w \cdot \delta_G$$

proporzionale sia al vantaggio della competenza sia all'incremento nella probabilità di acquisirla. Nel corso di generazioni, la frequenza dei genotipi favorevoli cresce, e con essa la probabilità media di acquisizione della competenza nella popolazione: la competenza diventa progressivamente più facile da apprendere, e il suo aspetto culturale si riduce mentre il suo aspetto biologico-architettonico cresce.

La differenza cruciale con la trasmissione lamarckiana — l'ereditarietà dei caratteri acquisiti — è che nell'effetto Baldwin la competenza stessa non si genetizza: ciò che si genetizza è la **facilità di acquisizione**. Il cervello diventa progressivamente meglio attrezzato per apprendere la competenza, non per eseguirla senza apprendimento. Questo produce un'architettura cognitiva che ha "pre-disposizioni" per certi tipi di apprendimento — una struttura di vincoli sull'apprendimento che il nativismo chomskiano interpretava come conoscenza innata, ma che l'effetto Baldwin suggerisce essere invece il risultato di millenni di co-evoluzione con pratiche culturali di apprendimento.

C.4 Il linguaggio come caso paradigmatico

La co-evoluzione gene-cultura dell'effetto Baldwin ha un caso paradigmatico nell'evoluzione del linguaggio, che illustra con precisione il meccanismo e le sue implicazioni per la questione del rapporto tra logica e intelligenza.

Il dibattito tra nativismo chomskiano e culturalismo è stato parzialmente dissolto dal riconoscimento che entrambe le posizioni colgono aspetti distinti della stessa realtà co-evolutiva. I bambini di tutte le culture acquisiscono il linguaggio con facilità straordinaria entro i primi tre anni di vita, anche in condizioni di input molto limitato — questo sembra supportare una predisposizione biologica. Ma le strutture grammaticali

specifiche variano enormemente tra le lingue del mondo, alcune ammettendo costruzioni che altre vietano radicalmente — questo sembra supportare il ruolo dell'apprendimento culturale.

La risoluzione co-evolutiva: l'evoluzione biologica ha selezionato per un cervello con architettura ottimizzata per l'acquisizione del linguaggio in generale — non per una grammatica universale nel senso tecnico chomskiano — e questa architettura è il risultato dell'effetto Baldwin che ha operato per decine o centinaia di migliaia di anni di pratiche linguistiche culturali. Il bambino nasce con un cervello le cui connessioni sinaptiche e la cui struttura corticale sono state modellate dall'effetto Baldwin per rendere l'acquisizione linguistica efficiente; le strutture grammaticali specifiche vengono apprese attraverso l'esposizione al linguaggio della comunità.

Questo modello ha una previsione verificabile e verificata: le strutture grammaticali che compaiono universalmente o quasi universalmente nelle lingue del mondo — le strutture per cui l'acquisizione è più rapida e robusta — dovrebbero essere quelle per cui il co-evolutive Baldwin effect ha avuto più tempo e più pressione selettiva per operare. Le strutture grammaticali rare o tipologicamente inusuali — quelle per cui l'acquisizione è più difficile e più dipendente dall'istruzione esplicita — dovrebbero corrispondere a innovazioni culturali recenti o a sistemi altamente specializzati.

L'implicazione per la logica formale è diretta. Il ragionamento sillogistico nella sua forma aristotelica, la logica proposizionale, la quantificazione del primo ordine: sono innovazioni culturali degli ultimi 2500 anni nella tradizione greco-europea, con paralleli in altre tradizioni (la logica *navya-nyāya* indiana, la logica cinese) che si sono sviluppate indipendentemente. L'effetto Baldwin ha avuto troppo poco tempo per

operare in modo significativo su queste pratiche: 2500 anni sono 100–125 generazioni, un tempo breve per la selezione genetica. L'architettura del cervello umano non è stata selezionata per il ragionamento sillogistico formale; è stata selezionata, nel corso di decine di migliaia di generazioni, per le pratiche cognitive che rendevano possibile la sopravvivenza e la cooperazione negli ambienti ancestrali.

C.5 Il problema del grounding semantico: la formulazione standard

Il problema del **radicamento semantico** (*symbol grounding problem*), formulato da Stevan Harnad nel 1990 nel saggio omonimo, è la questione di come i simboli — sequenze arbitrarie di glifo senza contenuto intrinseco — acquistino significato, ossia si connettano agli oggetti e agli stati del mondo che denotano. Nella sua formulazione originale, Harnad lo pone in questi termini: un dizionario cinese-cinese è inutile per chi non sa già il cinese, perché tutti i simboli sono definiti in termini di altri simboli. Il significato non emerge dalla relazione tra simboli: richiede una connessione con qualcosa al di fuori del sistema simbolico.

La risposta standard del funzionalismo computazionale — che il significato è determinato dalla struttura funzionale delle relazioni causali tra stati del sistema — è rigettata da Harnad con l'argomento che la struttura funzionale è essa stessa simbolica: è una relazione formale tra stati che non ha contenuto intrinseco. Il problema si pone a qualunque livello di astrazione: anche una descrizione funzionale completa di un sistema cognitivo è un insieme di simboli che non radicano il loro significato da soli.

La proposta di Harnad è che il grounding richieda due tipi di connessione con il mondo: il **grounding icònico** attraverso rappresentazioni analogiche delle caratteristiche percettive degli oggetti (il riconoscimento visivo di un gatto si radica nella struttura delle sue caratteristiche visive), e il **grounding categoriale** attraverso l'astrazione di caratteristiche invarianti tra istanze diverse della stessa categoria (il concetto *gatto* si radica nell'invariante estratto dall'esperienza percettiva di molti gatti diversi). Il linguaggio — il livello simbolico — si radica poi su queste rappresentazioni non-simboliche attraverso la connessione tra simboli e categorie percettive grounded.

Questa proposta ha il vantaggio di indicare un meccanismo concreto, ma ha due difficoltà: non specifica come avviene il grounding nei concetti astratti — i concetti matematici, i concetti normativi, i concetti teoretici della scienza — che non hanno connessione percettiva diretta; e non affronta la dimensione sociale del significato: il significato di *democrazia* o di *giustizia* non si radica nell'esperienza percettiva individuale ma nelle pratiche istituzionali della comunità.

C.6 Il grounding come minimizzazione dell'energia libera: la proposta di Friston

La prospettiva più ambiziosa sul problema del grounding semantico nei sistemi cognitivi è quella che emerge dalla teoria dell'**active inference** di Karl Friston, che estende la minimizzazione dell'energia libera variazionale dalla percezione all'azione e al significato. La proposta, sviluppata in collaborazione con filosofi come Jakob Hohwy e Chris Frith, è che il grounding semantico non sia una relazione statica tra simboli e referenti, ma un processo dinamico di costruzione di un **modello generativo** del mondo che minimizza l'energia libera.

Il punto di partenza è la formulazione dell'energia libera variazionale come funzione da minimizzare:

$$F = E_Q[\log Q(s) - \log P(o, s)]$$

dove $Q(s)$ è la distribuzione variazionale sugli stati nascosti s del mondo, $P(o, s)$ è il modello generativo del cervello — la distribuzione congiunta delle osservazioni o e degli stati nascosti — e l'aspettazione E_Q è rispetto alla distribuzione variazionale. Sviluppando:

$$F = E_Q[\log Q(s) - \log P(s|o) - \log P(o)] = D_{\text{KL}}(Q(s) \parallel P(s|o)) - \log P(o)$$

dove D_{KL} è la divergenza di Kullback-Leibler e $\log P(o)$ è il log-evidence delle osservazioni sotto il modello generativo. La divergenza KL è non-negativa, quindi $F \geq -\log P(o)$: l'energia libera è un limite superiore al negativo del log-evidence. Minimizzare F rispetto a Q equivale a minimizzare la divergenza tra la distribuzione variazionale e la posteriore reale $P(s|o)$, ossia a eseguire l'inferenza bayesiana nel modo più accurato possibile dati i vincoli computazionali.

L'estensione all'azione — l'**active inference** — modifica la formulazione includendo l'azione a come variabile del modello:

$$F(Q, a) = E_Q[\log Q(s) - \log P(o, s | a)]$$

Il sistema ora minimizza l'energia libera sia aggiornando le proprie credenze Q (percezione) sia scegliendo le azioni a che rendono le osservazioni future più probabili sotto il modello — ossia scegliendo le azioni che *confermano* le previsioni del modello. Questo produce un comportamento finalizzato non come massimizzazione di una ricompensa esterna, ma come minimizzazione della sorpresa attesa.

C.7 Il grounding come struttura del modello generativo

Nella prospettiva dell'active inference, il **significato** di un simbolo è la struttura del modello generativo che quel simbolo attiva: l'insieme delle previsioni che il cervello genera quando il simbolo è elaborato, le azioni che associa, gli stati affettivi e motivazionali che mobilita. Il grounding non è una relazione tra simbolo e referente: è la struttura causale del modello generativo che il simbolo parametrizza.

Questa formulazione dissolve la distinzione tra grounding icònico e grounding simbolico di Harnad: entrambi sono casi particolari della stessa relazione — la struttura del modello generativo associata a una rappresentazione. Il grounding percettivo è la struttura del modello generativo della modalità sensoriale pertinente: il concetto *rosso* attiva un modello generativo che predice pattern di attivazione nella corteccia visiva, forza risposte di orientamento, attiva aspettative di pericolo in certi contesti. Il grounding concettuale astratto è la struttura del modello generativo a livelli superiori della gerarchia: il concetto *giustizia* attiva un modello generativo di relazioni sociali, norme, aspettative di comportamento nelle istituzioni, che è costruito attraverso l'esperienza di interazioni sociali concrete.

Crucialmente, questa formulazione permette di trattare il grounding dei concetti matematici e logici in modo non misterioso. Il concetto *numero primo* attiva un modello generativo che predice il comportamento degli interi rispetto alla divisione, le distribuzioni dei gap tra primi, le proprietà degli algoritmi crittografici. Questo modello generativo è costruito attraverso l'esperienza di manipolazione di simboli matematici in pratiche culturali specifiche, ma è reale nel senso che produce previsioni verificabili e guida l'azione (la scelta di algoritmi, la costruzione di dimostrazioni). Il grounding matematico è funzionale-predittivo, non percettivo-diretto.

La distinzione tra sistemi simbolici e sistemi neurali acquista nuova precisione in questa prospettiva. I sistemi simbolici classici hanno una semantica designativa: i simboli si riferiscono a entità in un dominio per stipulazione (l'interpretazione I del modello formale). Questa semantica è esterna al sistema: è fornita dall'interprete umano che usa il sistema, non generata dal sistema stesso. I sistemi neurali, addestrati su corpora di testi prodotti da agenti che hanno modelli generativi ricchi del mondo, acquisiscono una semantica funzionale-predittiva: i loro stati interni attivano strutture di previsione che rispecchiano, anche se in modo approssimativo e indiretto, i modelli generativi degli agenti che hanno prodotto i testi. Il grounding è trasferito, non originario — ma questo non lo rende inesistente.

C.8 La misura dell'informazione come misura del grounding

La teoria dell'informazione fornisce uno strumento quantitativo per misurare il grounding semantico nella prospettiva dell'active inference. Se il grounding di un simbolo s è la struttura del modello generativo $P(o | s)$ che esso attiva — la distribuzione delle osservazioni previste — allora la ricchezza del grounding può essere misurata dall'**informazione mutua** tra il simbolo e le osservazioni:

$$I(S; O) = \sum_{s,o} P(s, o) \log [P(s, o) / (P(s) P(o))] = H(O) - H(O | S) = H(S) - H(S | O)$$

dove H è l'entropia di Shannon. L'informazione mutua misura quanto la conoscenza del simbolo riduce l'incertezza sulle osservazioni (e viceversa): è la misura del legame predittivo tra simbolo e mondo.

In questa prospettiva, un simbolo è *ben radicato* se l'informazione mutua tra esso e le osservazioni rilevanti è alta: il simbolo riduce sostanzialmente l'incertezza su ciò che si osserverà nel mondo. Un

simbolo è mal radicato se l'informazione mutua è bassa: il simbolo non predice nulla di specifico. I simboli dei sistemi esperti classici — le variabili proposizionali di MYCIN — hanno grounding solo nella misura in cui il progettista del sistema li ha connessi alle osservazioni cliniche attraverso le regole. Fuori dal dominio per cui le regole sono state scritte, il grounding si azzerava: le variabili perdono il loro potere predittivo.

I simboli nei modelli linguistici neurali hanno un grounding diverso: è distribuito, approssimativo, ma determinato statisticamente dalla co-occorrenza con descrizioni di osservazioni nel corpus. La parola *febbre* in un modello linguistico è radicata attraverso la sua co-occorrenza statisticamente strutturata con parole che descrivono sintomi, misurazioni, trattamenti — un grounding indiretto attraverso la co-occorrenza con il linguaggio di agenti che hanno grounding percettivo diretto. L'informazione mutua tra *febbre* e le sue co-occorrenze nel corpus è positiva e strutturata: il simbolo predice qualcosa sul contesto in cui appare.

Questa forma di grounding non è equivalente al grounding percettivo diretto di un agente incarnato. Ma non è nemmeno il grounding nullo di un sistema simbolico privo di connessioni con il mondo. È una forma di grounding derivato — la firma statistica delle relazioni tra simboli e mondo mediata attraverso il linguaggio umano — che produce competenze predittive reali pur senza la connessione causale diretta con il mondo che Harnad riteneva necessaria.

C.9 La co-evoluzione del linguaggio formale e del cervello

Le considerazioni delle sezioni precedenti convergono verso una tesi integrativa di notevole interesse: il rapporto tra le strutture logico-formali e il cervello umano non è statico ma co-evolutivo. Nel corso dei millenni

di pratica della logica, della matematica e della filosofia formale, l'effetto Baldwin ha presumibilmente operato — sia pure in modo debole per la brevità relativa del periodo — selezionando per cervelli con maggiore facilità di acquisizione del ragionamento formale astratto. Al tempo stesso, le pratiche formali si sono adattate alle architetture cognitive disponibili: i sistemi di notazione, i sistemi di prova, i sistemi pedagogici sono stati sviluppati per rendere il ragionamento formale acquisibile da cervelli che non erano architettate per esso.

Questa co-evoluzione tra strumento cognitivo e substrato cognitivo è un caso particolare di un fenomeno più generale che Merlin Donald, nel volume *Origins of the Modern Mind* del 1991, ha chiamato **exaptation cognitiva**: l'uso di capacità cognitive evoltesi per uno scopo per eseguire compiti radicalmente diversi che il processo evolutivo non aveva previsto. La neocorteccia è stata selezionata per la cognizione sociale, la navigazione spaziale, il riconoscimento di pattern visivi. La sua plasticità la rende disponibile per l'acquisizione di qualunque pratica culturale sufficientemente strutturata — compreso il ragionamento logico formale, la notazione musicale, la geometria euclidea.

Il risultato di questa exaptazione è che il cervello umano moderno — quello di un matematico che ha trascorso anni nello studio della logica formale — ha connessioni sinaptiche significativamente diverse da quelle di un individuo che non ha mai incontrato il pensiero formale. La pratica prolungata del ragionamento logico modifica strutturalmente il cervello: non nel senso lamarckiano di genetizzare la competenza, ma nel senso neuroplastico di rafforzare i circuiti che realizzano quella competenza. La logica lascia un'impronta nel cervello che la pratica.

Questa impronta neuroplastica è la manifestazione individuale di ciò che l'effetto Baldwin produce a livello di popolazione: un cervello che dopo

anni di pratica formale ha architetture sinaptiche ottimizzate per il ragionamento astratto. Ma è un'ottimizzazione individuale e reversibile, non una trasformazione genetica permanente: un bambino che non incontra mai il pensiero formale non svilupperà quelle connessioni, e un adulto che smette di praticare il ragionamento formale le perde progressivamente per disuso. La logica non è scritta nei geni: è scritta nella plasticità, che è molto più effimera dei geni ma anche molto più flessibile.

Espansione — Parte D

La filosofia della matematica pura dopo il doppio ridimensionamento, e la tesi finale integrata

D.1 Cosa rimane alla matematica dopo Gödel

I teoremi di incompletezza di Gödel hanno avuto un impatto traumatico sulla filosofia della matematica del XX secolo, ma l'elaborazione di quel trauma è stata disomogenea: alcune posizioni filosofiche sono state costrette a riconsiderarsi radicalmente, altre hanno trovato nel trauma una confutazione del rivale senza riconoscere le proprie fragilità. Prima di articolare le implicazioni della tesi di questa monografia per la filosofia della matematica pura, conviene fare un inventario preciso di ciò che i teoremi di Gödel hanno effettivamente distrutto e di ciò che hanno lasciato intatto.

Il **logicismo** di Frege e Russell — la tesi che la matematica sia riducibile alla logica pura — è stato colpito prima da Russell stesso con il paradosso del 1901 e poi consolidato dall'incompletezza: se la matematica fosse completamente catturata da un sistema logico, quel

sistema sarebbe completo, ma nessun sistema sufficientemente ricco è completo. Il logicismo non è però eliminato: sopravvive in forme indebolite come la tesi che la matematica abbia struttura logica, non che la matematica *sia* logica.

Il **formalismo hilbertiano** — nella sua versione forte della derivabilità di tutta la matematica da un sistema di assiomi finito e dimostrabilmente coerente con metodi finitari — è stato confutato direttamente. Ma il programma formale come prassi matematica — costruire sistemi assiomatici rigorosi, verificare la coerenza relativa, studiare le proprietà dei sistemi formali — è sopravvissuto e ha prosperato in teoria dei modelli, teoria della dimostrazione, teoria degli insiemi.

Il **intuizionismo** di Brouwer — la tesi che la matematica sia un'attività mentale costruttiva e che esistano solo gli oggetti matematici che possono essere costruiti mentalmente — ha ricevuto un paradossale supporto dall'incompletezza: se esistono proposizioni indecidibili, forse la matematica non è uno spazio di oggetti platonici pre-esistenti ma una costruzione. Ma l'intuizionismo porta a conseguenze che la maggior parte dei matematici trova inaccettabili: il rifiuto del principio del terzo escluso, l'abbandono della dimostrazione per assurdo non costruttiva, la rinuncia a gran parte dell'analisi classica.

Il **platonismo matematico** — la tesi che gli oggetti matematici esistano indipendentemente dalla mente — sopravvive all'incompletezza in modo notevole. Gödel stesso era un platonista convinto: la formula G è vera nell'aritmetica standard pur essendo indimostrabile nel sistema, il che suggerisce che la verità matematica eccede la dimostrabilità formale e deve fare riferimento a qualcosa di indipendente dal sistema. Ma il platonismo solleva il problema epistemologico acuto di Paul Benacerraf (1973): se gli oggetti matematici esistono in un dominio astratto privo di

proprietà causali, come è possibile la conoscenza matematica? Come può il cervello fisico di un matematico entrare in contatto causale con oggetti astratti per conoscerli?

D.2 Il naturalismo matematico e la tesi di Maddy

Una posizione filosofica sulla matematica che si adatta bene alla prospettiva di questa monografia è il **naturalismo matematico** di Penelope Maddy, sviluppato nei volumi *Realism in Mathematics* (1990) e *Naturalism in Mathematics* (1997). Maddy propone di trattare la matematica come una pratica umana autonoma e di spiegare i suoi metodi e i suoi standard non per riferimento a oggetti platonici, ma attraverso un'analisi naturalistica della pratica matematica stessa.

Il punto di partenza di Maddy è che la questione "perché i matematici accettano certi assiomi e non altri?" ha una risposta pragmatica, non platonica: i matematici accettano gli assiomi dello ZFC perché sono utili per la matematica corrente, permettono di derivare i teoremi che si vuole derivare, e non producono contraddizioni con risultati consolidati. La domanda filosofica "ma quegli assiomi sono *veri*?" è mal posta nella sua forma assoluta: la verità degli assiomi è la loro fertilità matematica.

Per gli assiomi del transfinito — gli assiomi che vanno al di là del necessario per la matematica corrente, come i grandi cardinali — Maddy propone di valutarli attraverso ciò che chiama *good set theory*: gli assiomi che permettono di rispondere alle domande matematiche che rimangono indecidibili in ZFC, che producono teorie gerarchicamente coerenti, e che generalizzano in modo naturale gli assiomi già accettati. Questo è un criterio pragmatico e coerente con la pratica matematica effettiva.

La connessione con la tesi di questa monografia è la seguente: il naturalismo di Maddy tratta la matematica come una pratica cognitiva

umana soggetta a vincoli pratici — la derivabilità, l'utilità, la coerenza con la pratica corrente — non come la scoperta di verità platoniche pre-esistenti. Questo è coerente con la tesi che la competenza matematica sia una tecnologia cognitiva costruita su una base non-logica: la pratica matematica non è l'esplorazione di un dominio astratto pre-esistente, ma la costruzione progressiva di strutture formali che soddisfano certi criteri pratici di fertilità e coerenza.

D.3 La matematica come ottimizzazione: il programma di ricerca

La tesi più originale che questa monografia può proporre per la filosofia della matematica pura è la seguente: la pratica matematica è essa stessa un processo di ottimizzazione, strutturalmente analogo — anche se non identico — al processo di addestramento dei sistemi neurali artificiali.

Il matematico che esplora un problema riceve segnali di feedback dalla struttura matematica: la difficoltà di una dimostrazione, l'eleganza di una costruzione, la naturalezza di una generalizzazione, la fertilità di un assioma nel produrre teoremi interessanti. Questi segnali guidano l'esplorazione dello spazio dei concetti e delle costruzioni matematiche, favorendo alcune direzioni e scoraggiandone altre. Il risultato — i teoremi, le strutture, i concetti che sopravvivono e vengono trasmessi attraverso le generazioni di matematici — è l'output di un processo di selezione che opera sullo spazio delle possibilità matematiche.

Questo processo di selezione ha proprietà formalmente simili all'ottimizzazione per gradient descent. Il gradiente del processo matematico è il segnale di fertilità: la direzione nello spazio dei concetti che produce più risultati con meno risorse. Il tasso di apprendimento è modulato dalla difficoltà del problema: problemi molto duri vengono

esplorati lentamente con passi piccoli, problemi accessibili vengono risolti rapidamente con passi grandi. Il fenomeno della doppia discesa ha un analogo matematico: periodi di apparente complessità e confusione (l'inverno di un'area matematica, quando non si vedono progressi) precedono spesso breakthrough che semplificano radicalmente il panorama, corrispondendo all'uscita dalla regione di interpolazione critica.

Questa analogia non è una riduzione della matematica a un algoritmo di ottimizzazione: è un'osservazione strutturale sulla dinamica della pratica matematica. Il valore del matematico — la sua creatività, la sua intuizione, la sua capacità di scegliere le direzioni fruttuose — non è eliminato dall'analogia: è precisamente ciò che rende efficiente il processo di ottimizzazione, così come le buone euristiche rendono efficiente il gradient descent su superfici complesse.

D.4 L'indipendenza delle proposizioni gödeliane e la pratica matematica

La questione più concreta che la filosofia della matematica deve affrontare dopo Gödel è: cosa fare delle proposizioni indecidibili? L'incompletezza non è una curiosità esotica: produce risultati di indecidibilità che toccano matematica corrente.

Il teorema di indipendenza di Cohen (1963), che ha completato il lavoro di Gödel sulla coerenza dell'ipotesi del continuo (CH) del 1938, mostra che CH è indipendente dagli assiomi ZFC: né CH né $\neg$ CH sono deducibili dagli assiomi standard della teoria degli insiemi. L'ipotesi del continuo afferma che non esiste un insieme di cardinalità strettamente compresa tra quella dei naturali $\aleph_0$ e quella dei reali $2^{\aleph_0}$. Il risultato di

Gödel-Cohen mostra che entrambe le posizioni sono matematicamente coerenti.

La posizione di Gödel era che l'indipendenza fosse un segnale di inadeguatezza degli assiomi di ZFC, e che i matematici dovessero cercare nuovi assiomi che decidessero CH in modo naturale. Il suo candidato erano gli **assiomi di grandi cardinali** — assiomi che affermano l'esistenza di cardinali inaccessibili, Mahlo, misurabili, supercompatti — che estendono la gerarchia degli insiemi transfiniti in modi progressivamente più ricchi. Il programma di Gödel-Woodin per decidere CH attraverso i grandi cardinali è un'area di ricerca attiva: i risultati di Woodin suggeriscono che CH sia falso in una teoria naturale con assiomi di grandi cardinali, ma la questione rimane aperta e controversa.

La posizione alternativa del **pluralismo matematico** — difesa da Joel Hamkins e altri — è che le diverse teorie degli insiemi (ZFC+CH, ZFC+¬CH, ZFC+assiomi di grandi cardinali) siano tutte ugualmente legittime e corrispondano a diversi "universi" matematici. Non c'è un unico insieme vero: ci sono molti insiemi diversi, tutti matematicamente corretti rispetto ai propri assiomi. Questa posizione ha la conseguenza radicale che la matematica degli insiemi non è la scoperta di una struttura pre-esistente ma la costruzione di molte strutture alternative.

Per la tesi di questa monografia, entrambe le posizioni sono coerenti. Il pluralismo di Hamkins è la versione insiemistica del naturalismo di Maddy: la matematica è una pratica che costruisce strutture secondo criteri di coerenza e fertilità, e non c'è ragione di privilegiare a priori una struttura sulle altre. La posizione di Gödel-Woodin è la versione platonizzante: esiste una struttura vera dell'universo degli insiemi, e i nuovi assiomi devono catturarla. La scelta tra le due posizioni — e non è

chiaro se questa scelta sia scientificamente decidibile — non cambia la tesi centrale di questa monografia, che riguarda il livello della pratica cognitiva, non il livello ontologico.

D.5 Il teorema di incompletezza come risorsa, non come limite

Un punto che la vulgata sulla incompletezza di Gödel sistematicamente fraintende è la natura della relazione tra i teoremi di Gödel e la pratica matematica corrente. I teoremi dimostrano l'esistenza di proposizioni indecidibili — ma nella stragrande maggioranza della matematica corrente, l'ind decidibilità non si manifesta. I teoremi dell'analisi, dell'algebra, della geometria, della topologia differenziale: sono quasi tutti decidibili in ZFC o nelle sue estensioni naturali. L'incompletezza morde ai bordi — nelle proposizioni della teoria degli insiemi di alta complessità — non al centro della pratica matematica quotidiana.

Più importante: l'incompletezza ha prodotto una **risorsa** per la matematica — la teoria della dimostrazione come disciplina autonoma — che è di per sé uno degli sviluppi più fecondi della matematica del XX secolo. Lo studio delle dimostrazioni come oggetti matematici, la classificazione dei principi dimostrativi per forza (PA , ACA_0 , $\Pi^1_1-CA_0$ nella gerarchia dei sistemi del **reverse mathematics** di Harvey Friedman e Stephen Simpson), la connessione tra la forza dimostrativa di un sistema e i tipi di funzioni che può dimostrare totali (la corrispondenza di Curry-Howard tra tipi e proposizioni): tutto questo è emerso dal programma gödeliano e ha prodotto matematica profonda che nulla avrebbe a che fare con il collasso del fondazionalismo.

Il **reverse mathematics** di Friedman e Simpson merita attenzione particolare perché illustra come l'incompletezza sia diventata uno

strumento analitico preciso. Il programma studia, per ogni teorema matematico ordinario (un teorema dell'analisi reale, dell'algebra, della combinatoria), quale sistema assiomatico è necessario e sufficiente per dimostrarlo. Il risultato principale è che quasi tutti i teoremi dell'analisi ordinaria cadono in uno di cinque sistemi assiomatici ("the Big Five"), ordinate per forza dimostrativa crescente: RCA_0 , WKL_0 , ACA_0 , ATR_0 , $\Pi^1_1-CA_0$. Questo schema rivela la struttura gerarchica della forza dimostrativa nascosta nei teoremi ordinari: il teorema di Bolzano-Weierstrass richiede ACA_0 , il teorema di Ramsey per gli insiemi infiniti richiede sistemi più forti di qualunque sistema compattato in una teoria del secondo ordine.

D.6 La computabilità come struttura dell'intelligenza matematica

La connessione più profonda tra la teoria della computabilità e la matematica pura è quella stabilita dalla **corrispondenza di Curry-Howard**, che identifica proposizioni logiche con tipi di programmazione e dimostrazioni con programmi. In questa corrispondenza:

una proposizione P corrisponde a un tipo $\tau(P)$ una dimostrazione di P corrisponde a un termine t di tipo $\tau(P)$ la verifica della correttezza di una dimostrazione corrisponde al type-checking di t

Le regole di inferenza logica corrispondono a costruzioni sui tipi:

l'introduzione dell'implicazione $A \rightarrow B$ corrisponde all'astrazione funzionale $\lambda x.t$ di tipo $A \rightarrow B$ l'eliminazione dell'implicazione (modus ponens) corrisponde all'applicazione funzionale $t u$ l'introduzione della congiunzione $A \wedge B$ corrisponde alla costruzione di coppia $\langle t, u \rangle$ di tipo $A \times B$ l'introduzione della disgiunzione $A \vee B$ corrisponde alle iniezioni

$\text{inl}(t)$ e $\text{inr}(u)$ la quantificazione universale $\forall x.P(x)$ corrisponde alla dipendenza di tipo $\Pi_{x:A} \tau(P(x))$ la quantificazione esistenziale $\exists x.P(x)$ corrisponde al tipo dipendente $\Sigma_{x:A} \tau(P(x))$

Questa corrispondenza ha implicazioni filosofiche di grande portata: dimostrare un teorema è equivalente a costruire un programma. Un sistema di verifica formale — un **proof assistant** come Coq, Lean, o Agda — è un type-checker per un linguaggio in cui le proposizioni matematiche sono tipi e le dimostrazioni sono programmi di quei tipi. Il recente progetto di formalizzazione in Lean della dimostrazione del teorema di Fermat — condotto da Kevin Buzzard e collaboratori — illustra la potenza operativa di questa corrispondenza: migliaia di pagine di matematica profonda vengono ridotte a un programma di tipo verificabile meccanicamente.

La corrispondenza di Curry-Howard mostra che la logica e la computazione non sono due sistemi indipendenti: sono due aspetti dello stesso oggetto matematico. Le strutture che permettono di esprimere le dimostrazioni — i tipi dipendenti, la teoria omotopica dei tipi (HoTT) — sono le stesse strutture che permettono di esprimere i programmi. Questo non significa che la logica sia primitiva rispetto alla computazione o viceversa: significa che entrambe sono facce di una struttura matematica più profonda.

D.7 La homotopy type theory come nuova fondazione

Il progetto più ambizioso nell'intersezione tra logica, computazione e matematica pura degli ultimi vent'anni è la **Homotopy Type Theory** (HoTT), sviluppata a partire dalle connessioni tra la teoria dei tipi di Martin-Löf e la teoria dell'omotopia algebrica scoperte da Voevodsky, Awodey, Warren e altri. Il volume *Homotopy Type Theory: Univalent*

Foundations of Mathematics (2013), frutto di un anno di lavoro collaborativo all'Institute for Advanced Study di Princeton, propone una nuova fondazione per la matematica in cui i tipi sono trattati come spazi topologici e i termini dello stesso tipo come punti dello stesso spazio.

Il concetto chiave è il **principio di univocità** (*univalence axiom*) di Voevodsky: due tipi A e B sono uguali ($A =_U B$ nell'universo U) se e solo se sono equivalenti come spazi omotopici ($A \simeq B$). Formalmente, l'assioma afferma che la mappa naturale:

$$(A = B) \rightarrow (A \simeq B)$$

è essa stessa un'equivalenza. Questo significa che isomorfismo implica identità: strutture matematiche equivalenti non sono solo isomorfe ma *identiche* nell'universo HoTT . La distinzione standard tra "uguale a" e "isomorfo a" che pervade la matematica classica — e che produce una proliferazione di proposizioni del tipo "a meno di isomorfismo" — viene eliminata: le strutture equivalenti sono la stessa struttura.

HoTT ha la proprietà di essere costruttiva — ogni teorema ha un termine computazionale associato — e di incorporare la teoria dell'omotopia come struttura fondazionale piuttosto che come conseguenza. La gerarchia n -trunca dei tipi rispecchia la gerarchia dei gruppi di omotopia: i tipi di livello (-1) sono le proposizioni (hanno al più un termine), i tipi di livello 0 sono gli insiemi (i loro termini hanno al più un percorso tra loro), i tipi di livello 1 sono i gruppidi (i cui percorsi formano gruppi), e così via.

Per la tesi di questa monografia, HoTT è un caso particolarmente istruttivo: rappresenta un tentativo sofisticato di ricostituire le fondamenta della matematica non su un sistema logico assiomatico classico (come ZFC) ma su una teoria dei tipi costruttiva con struttura omotopica. Questo tentativo è il riconoscimento implicito che la logica

classica del primo ordine — la logica dello ZFC — non è l'unica fondazione possibile per la matematica, e che strutture computazionali più ricche possono fornire fondazioni altrettanto solide e più espressive. È, in questo senso, un movimento nella direzione opposta al fondazionalismo hilbertiano: invece di ridurre la matematica alla logica più semplice, espande le fondazioni verso strutture computazionali e geometriche più ricche.

D.8 La tesi finale integrata: cinque affermazioni che resistono alla critica

Siamo ora in posizione di formulare, con la precisione che l'intera argomentazione ha costruito, la tesi finale integrata della monografia e delle sue espansioni. La tesi si articola in cinque affermazioni distinte, a gradi diversi di generalità, ciascuna difendibile sulla base degli argomenti sviluppati.

La **prima affermazione** è matematica e non contestabile: i sistemi formali sufficientemente ricchi da contenere l'aritmetica elementare sono necessariamente incompleti (Gödel) e nessun algoritmo può decidere la terminazione di un programma arbitrario (Turing). Questi risultati non sono limiti tecnici ma proprietà strutturali dello spazio della computazione e della dimostrazione, derivate da argomenti diagonali che si riproducono a qualunque livello di potenza del sistema.

La **seconda affermazione** è storico-ingegneristica e ben documentata: i sistemi di intelligenza artificiale basati su rappresentazioni simboliche esplicite e inferenza logica formale hanno fallito nel generalizzare al dominio aperto per ragioni strutturali — il frame problem, il qualification problem, il ramification problem — che non si lasciano risolvere per raffinamento tecnico. Il collasso non è un incidente storico:

è la manifestazione operativa degli stessi limiti strutturali della rappresentazione formale completa.

La **terza affermazione** è computazionale e supportata da derivazioni matematiche esplicite: i sistemi neurali profondi addestrati per minimizzazione della cross-entropia approssimano il prior universale di Solomonoff sulla distribuzione di addestramento, e le loro proprietà di generalizzazione sono governate dalla dimensione VC, dalla complessità di Rademacher, e dalla regolarizzazione implicita del gradient descent stocastico. La competenza cognitiva che esibiscono è una proprietà emergente della distribuzione appresa, non il prodotto di un sistema di regole esplicite.

La **quarta affermazione** è biologica e difendibile con le cautele appropriate: il cervello umano è il prodotto di ottimizzazione evolutiva per la sopravvivenza in ambienti aperti, e i suoi meccanismi fondamentali — plasticità hebbiana, codifica predittiva gerarchica, inferenza bayesiana approssimata — non sono meccanismi logici. La capacità di ragionamento logico formale è una competenza acquisita attraverso pratiche culturali, con un contributo dell'effetto Baldwin che è reale ma minore per la brevità evolutiva del periodo di pratica formale. Il grounding semantico delle rappresentazioni — sia biologiche sia artificiali — è meglio descritto come struttura del modello generativo che minimizza l'energia libera, non come connessione designativa tra simbolo e referente.

La **quinta affermazione** è filosofica e costituisce il contributo più originale: la logica formale non è il fondamento dell'intelligenza ma uno dei suoi strumenti più elevati, e questo riposizionamento non diminuisce il suo valore nei domini per cui è adeguata ma ridimensiona le sue pretese fondazionali. Le due crisi della logica — la crisi interna

dell'incompletezza e la crisi operativa dell'AI simbolica — non sono eventi separati: sono manifestazioni della stessa scoperta vista da angolature diverse. La struttura aperta del mondo — la texture aperta di Waismann — non è catturabile da nessun sistema formale finito, né nella matematica né nell'intelligenza pratica. E l'intelligenza, biologica e artificiale, prospera precisamente nella sua capacità di navigare questa apertura attraverso l'approssimazione ottimale, non nonostante di essa.

D.9 Le questioni che rimangono aperte

La completezza intellettuale impone di elencare con onestà le questioni che la monografia non ha risolto e che rimangono genuinamente aperte.

La prima questione aperta riguarda la **coscienza e la comprensione genuina**. La tesi che i sistemi neurali realizzino competenza cognitiva senza implementare un motore logico non dice nulla sulla questione se quei sistemi abbiano *comprensione* nel senso fenomenologico — se vi sia qualcosa che si prova ad essere un modello linguistico. Il problema difficile della coscienza di Chalmers rimane aperto, e la tesi di questa monografia è compatibile sia con una risposta positiva sia con una negativa.

La seconda questione aperta riguarda la **scalabilità dell'approssimazione bayesiana**. I sistemi neurali attuali approssimano il prior universale di Solomonoff sulla distribuzione dei testi umani. Non è chiaro se questa approssimazione sia sufficiente per tutti i tipi di ragionamento — in particolare per il ragionamento matematico profondo che richiede catene inferenziali molto lunghe e la costruzione di oggetti astratti radicalmente nuovi. Le leggi di scala suggeriscono che la qualità dell'approssimazione migliori monotonamente con le risorse,

ma non escludono che esistano tipi di ragionamento che richiedono risorse qualitativamente diverse.

La terza questione aperta riguarda il **fondamento delle matematiche**. Il pluralismo di Hamkins e il monismo di Gödel-Woodin sono entrambi coerenti con la tesi di questa monografia, e la scelta tra essi dipende da domande ontologiche sulla natura degli oggetti matematici che questa monografia ha deliberatamente lasciato aperte. HoTT offre una via alternativa che reintegra computazione e matematica a un livello più profondo, ma la sua adozione come fondazione universale è ancora in corso di valutazione dalla comunità matematica.

La quarta questione aperta riguarda la **co-evoluzione futura**. Se i sistemi neurali artificiali diventano strumenti cognitivi pervasivi — e le evidenze suggeriscono che lo stiano diventando — l'effetto Baldwin opererà anche sull'interazione tra cervelli umani e sistemi artificiali: i cervelli umani verranno selezionati per la facilità di collaborazione con questi sistemi, e i sistemi verranno ottimizzati per le architetture cognitive umane. La direzione di questa co-evoluzione — e le sue implicazioni per le capacità cognitive umane — è genuinamente imprevedibile sulla base delle conoscenze attuali.

Nota bibliografica dell'espansione

Per la teoria della complessità algoritmica: Kolmogorov, *Three Approaches to the Quantitative Definition of Information*, in *Problems of Information Transmission*, 1 (1965); Chaitin, *On the Length of Programs for Computing Finite Binary Sequences*, in *Journal of the ACM*, 13 (1966); Solomonoff, *A Formal Theory of Inductive Inference, Parts I and II*, in *Information and Control*, 7 (1964); Rissanen, *Modeling by Shortest Data Description*, in *Automatica*, 14 (1978).

Per la teoria PAC e la generalizzazione: Valiant, *A Theory of the Learnable*, in *Communications of the ACM*, 27 (1984); Vapnik e Chervonenkis, *On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities*, in *Theory of Probability and Its Applications*, 16 (1971); Bartlett e Mendelson, *Rademacher and Gaussian Complexities: Risk Bounds and Structural Results*, in *Journal of Machine Learning Research*, 3 (2002); Belkin, Hsu, Ma e Mandal, *Reconciling Modern Machine-Learning Practice and the Classical Bias-Variance Trade-Off*, in *Proceedings of the National Academy of Sciences*, 116 (2019).

Per l'in-context learning: Xie, Raghunathan, Liang e Ma, *An Explanation of In-Context Learning as Implicit Bayesian Inference*, in *Proceedings of ICLR* (2022); Akyürek, Schuurmans, Andreas, Ma e Zhou, *What Learning Algorithm is In-Context Learning? Investigations with Linear Models*, in *Proceedings of ICLR* (2023); Brown, Mann, Ryder e colleghi, *Language Models are Few-Shot Learners (GPT-3)*, in *Proceedings of NeurIPS* (2020).

Per le leggi di scala: Kaplan, McCandlish, Henighan, Brown, Chess, Child, Gray, Radford, Wu e Amodei, *Scaling Laws for Neural Language Models*, arXiv:2001.08361 (2020).

Per la co-evoluzione gene-cultura: Cavalli-Sforza e Feldman, *Cultural Transmission and Evolution: A Quantitative Approach* (1981); Boyd e Richerson, *Culture and the Evolutionary Process* (1985); Baldwin, *A New Factor in Evolution*, in *The American Naturalist*, 30 (1896); Donald, *Origins of the Modern Mind: Three Stages in the Evolution of Culture and Cognition* (1991).

Per il grounding semantico e l'active inference: Harnad, *The Symbol Grounding Problem*, in *Physica D*, 42 (1990); Friston, *The Free-Energy Principle: A Unified Brain Theory?*, in *Nature Reviews Neuroscience*, 11 (2010); Friston, FitzGerald, Rigoli, Schwartenbeck e Pezzulo, *Active Inference: A Process Theory*, in *Neural Computation*, 29 (2017); Hohwy, *The Predictive Mind*

(2013); Friston, *A Free Energy Principle for a Particular Physics*, arXiv:1906.10184 (2019).

Per la filosofia della matematica: Benacerraf, *Mathematical Truth*, in *Journal of Philosophy*, 70 (1973); Gödel, *What is Cantor's Continuum Problem?*, in *American Mathematical Monthly*, 54 (1947); Cohen, *The Independence of the Continuum Hypothesis*, in *Proceedings of the National Academy of Sciences*, 50 (1963); Maddy, *Realism in Mathematics* (1990) e *Naturalism in Mathematics* (1997); Hamkins, *The Set-Theoretic Multiverse*, in *Review of Symbolic Logic*, 5 (2012); Woodin, *The Continuum Hypothesis, Part I*, in *Notices of the AMS*, 48 (2001); Friedman e Simpson, *Issues and Problems in Reverse Mathematics*, in *Computability Theory and Its Applications*, Contemporary Mathematics, 257 (2000); Voevodsky e colleghi, *Homotopy Type Theory: Univalent Foundations of Mathematics* (2013); Curry, *Functionality in Combinatory Logic*, in *Proceedings of the National Academy of Sciences*, 20 (1934); Howard, *The Formulae-as-Types Notion of Construction*, manoscritto del 1969, pubblicato in *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism* (1980).

Per la filosofia della mente e il funzionalismo: Searle, *Minds, Brains, and Programs*, in *Behavioral and Brain Sciences*, 3 (1980); Putnam, *Psychological Predicates*, in *Art, Mind, and Religion*, a cura di W.H. Capitan e D.D. Merrill (1967); Benacerraf, *What Numbers Could Not Be*, in *Philosophical Review*, 74 (1965); Chalmers, *The Conscious Mind: In Search of a Fundamental Theory* (1996); Quine, *Two Dogmas of Empiricism*, in *Philosophical Review*, 60 (1951); Goodman, *Fact, Fiction, and Forecast* (1955); Carnap, *Der logische Aufbau der Welt* (1928).

Per il Circolo di Vienna: Neurath, Carnap e Hahn, *Wissenschaftliche Weltauffassung: Der Wiener Kreis* (1929); Schlick, *Allgemeine Erkenntnislehre* (1918); Waismann, *Verifiability*, in *Proceedings of the Aristotelian Society*, supplementary vol. 19 (1945).

Espansione — Parte E

P versus NP, l'asimmetria verifica-scoperta e l'epistemologia della creatività matematica

E.1 La domanda più profonda della teoria della computazione

C'è una questione che la teoria della computazione ha sollevato e che, pur essendo formulata in termini tecnici di apparente sobrietà, ha implicazioni per la natura dell'intelligenza e della conoscenza forse più profonde dei teoremi di Gödel e Turing. Questa questione è il problema P versus NP, formulato indipendentemente da Cook (1971) e Levin (1973), e rimasto irrisolto da oltre cinquant'anni nonostante l'attenzione intensa di centinaia di matematici e informatici di primo livello. La sua irrisolubilità non è casuale: tocca qualcosa di strutturale sulla natura del ragionamento e della scoperta.

La formulazione tecnica è la seguente. La classe P è l'insieme dei problemi decisionali risolvibili in tempo polinomiale da una macchina di Turing deterministica: problemi per cui esiste un algoritmo con complessità $O(n^k)$ per qualche costante k, dove n è la dimensione dell'input. La classe NP è l'insieme dei problemi decisionali per cui una soluzione proposta può essere *verificata* in tempo polinomiale: dato un "certificato" c e un input x, si può controllare in tempo $O(|x|^k)$ se c è una soluzione valida per x.

$P \subseteq NP$ per definizione: se un problema è risolvibile in tempo polinomiale, allora la soluzione è certamente verificabile in tempo polinomiale (basta rieseguire l'algoritmo). La domanda $P = NP$? chiede se valga anche la direzione inversa: se tutti i problemi facilmente verificabili sono anche facilmente risolvibili. Il consenso tra i teorici è che

$P \neq NP$, ma la dimostrazione rimane uno dei sette Problemi del Millennio del Clay Mathematics Institute con un premio di un milione di dollari.

E.2 NP-completezza e la struttura della difficoltà computazionale

Il teorema di Cook (1971) e il teorema di Karp (1972) hanno mostrato che NP ha una struttura interna precisa: esistono problemi **NP-completi** — problemi in NP tale che ogni problema in NP si riduce ad essi in tempo polinomiale. Se uno qualunque dei problemi NP-completi fosse risolvibile in tempo polinomiale, allora $P = NP$. Questi problemi sono, in un senso preciso, i più difficili problemi in NP.

Il problema NP-completo originale di Cook è la soddisfacibilità proposizionale (SAT): data una formula proposizionale in forma normale congiuntiva (CNF) — una congiunzione di clausole, ciascuna disgiunzione di letterali — esiste un'assegnazione di verità alle variabili che soddisfi tutte le clausole? La riduzione di Cook mostra che qualunque problema in NP può essere trascritto in un'istanza di SAT in tempo polinomiale: SAT è il problema universale della verifica computazionale.

Il teorema di Karp ha mostrato che 21 problemi classici di ottimizzazione combinatoria — il problema del commesso viaggiatore (nella versione di decisione), il problema della clique massima in un grafo, la colorazione dei grafi, lo zaino, la copertura dei vertici — sono tutti NP-completi e dunque tutti equivalenti in difficoltà computazionale. Questo è un risultato di grande eleganza strutturale: problemi che sembrano completamente diversi — pianificazione di percorsi,

partizionamento di risorse, assegnazione di compiti — hanno tutti la stessa difficoltà intrinseca.

La riduzione polinomiale tra problemi definisce una struttura di ordine parziale sulla classe NP — la **gerarchia di Karp-Levin** — che generalizza la gerarchia aritmetica di Kleene alle domande di complessità computazionale polinomiale. Nella gerarchia polinomiale, i problemi sono classificati per livelli Σ^P e Π^P a seconda della struttura dei quantificatori alternanti necessari per definirli: $\Sigma_1^P = \text{NP}$ ($\exists$ soluzione verificabile in P), $\Pi_1^P = \text{co-NP}$ ($\forall$ assegnazioni la formula è insoddisfatta), Σ_2^P ($\exists \dots \forall \dots$), e così via. Se $P = \text{NP}$, la gerarchia collassa completamente: tutti i livelli collassano a P. La congettura $P \neq \text{NP}$ è equivalente all'ipotesi che la gerarchia non collassi.

E.3 L'asimmetria verifica-scoperta come fenomeno cognitivo fondamentale

L'importanza del problema P vs NP per la tesi di questa monografia non è solo tecnica. C'è un'asimmetria profonda e universale nell'esperienza cognitiva umana che il problema cattura formalmente: **verificare una soluzione è sistematicamente più facile che trovarla.**

Questa asimmetria pervade la cognizione a tutti i livelli. In matematica, verificare la correttezza di una dimostrazione di 100 pagine richiede competenza e attenzione, ma è concettualmente diverso dal trovare quella dimostrazione: Wiles ha impiegato sette anni a dimostrare il teorema di Fermat, ma un matematico competente può verificare la dimostrazione in settimane. In musica, un ascoltatore educato può riconoscere la bellezza e la coerenza di una composizione di Bach, ma non potrebbe comporla. In linguaggio, capire una frase è più facile che produrla — i bambini capiscono strutture grammaticali molto più

complesse di quelle che producono. In scienza, valutare la plausibilità di una teoria è più facile che inventarla.

Se $P = NP$, questa asimmetria sarebbe un'illusione: tutto ciò che è verificabile sarebbe trovabile con lo stesso costo computazionale, e la creatività — la scoperta di nuove soluzioni — non richiederebbe nulla di più della verifica. Ogni problema che si lascia riconoscere quando risolto si lascerebbe risolvere con la stessa facilità. La matematica diventerebbe un'attività meccanica: scrivere un programma che trova dimostrazioni sarebbe altrettanto facile che scrivere un programma che le verifica. Le sinfonie si compongono automaticamente nella misura in cui si può riconoscere una sinfonia riuscita.

Se $P \neq NP$ — la congettura prevalente — l'asimmetria è reale e strutturale: esistono problemi per cui la scoperta è esponenzialmente più costosa della verifica, indipendentemente dall'algoritmo usato. La creatività matematica — e più in generale la capacità di trovare soluzioni a problemi difficili — non è riducibile alla verifica sistematica delle possibilità.

E.4 Il contenuto epistemico di $P \neq NP$

La congettura $P \neq NP$, se vera, ha implicazioni epistemiche di prima grandezza che la discussione tecnica lascia spesso implicite.

La prima implicazione riguarda la **natura della dimostrazione matematica**. I teoremi matematici profondi — Fermat, Poincaré, la classificazione dei gruppi finiti semplici — non sono stati trovati cercando sistematicamente nello spazio delle possibili dimostrazioni, ma attraverso intuizioni, analogie, salti creativi che hanno ridotto radicalmente lo spazio di ricerca. Se $P \neq NP$, questo non è un difetto dei matematici: è la struttura del problema. Trovare una dimostrazione di

lunghezza n non è più facile — asintoticamente — di verificare tutte le 2^n possibili dimostrazioni di quella lunghezza. L'intuizione matematica non è solo un aiuto pratico: è computazionalmente necessaria.

La seconda implicazione riguarda il rapporto tra **intelligenza e algoritmo**. Se $P \neq NP$, nessun algoritmo può risolvere sistematicamente i problemi NP-completi in tempo polinomiale. Ma i sistemi cognitivi — biologici e artificiali — *sembrano* risolvere (o approssimare) questi problemi in modo efficiente in molti casi pratici. Come? La risposta è che i problemi pratici non sono istanze pessime: hanno struttura che gli algoritmi esatti non sfruttano ma che le euristiche e i sistemi apprendenti riconoscono. Il cervello e le reti neurali non risolvono il problema generale: risolvono la distribuzione di problemi che si presentano nel dominio di addestramento, sfruttando la struttura statistica di quella distribuzione.

Questo è precisamente il meccanismo della connessione tra la complessità computazionale e l'apprendimento statistico che questa monografia ha sviluppato: i sistemi neurali non violano $P \neq NP$, ma la aggirano sfruttando la struttura della distribuzione dei problemi reali. La congettura $P \neq NP$ pone un limite su cosa possono fare tutti gli algoritmi, ma l'apprendimento statistico dice qualcosa di più fine: le distribuzioni di problemi reali hanno struttura che rende possibile l'approssimazione efficiente anche quando il problema generale è difficile.

E.5 La dimostrazione dell'NP-completezza di SAT: il teorema di Cook

Poiché il teorema di Cook è il fondamento dell'intera teoria dell'NP-completezza, vale la pena delineare la struttura della dimostrazione con sufficiente dettaglio per apprezzare la tecnica.

Il teorema afferma che SAT è NP-difficile: ogni problema in NP si riduce a SAT in tempo polinomiale. La dimostrazione costruisce, per ogni problema L in NP e ogni input x , una formula $\varphi_{\{L,x\}}$ in CNF tale che $\varphi_{\{L,x\}}$ è soddisfacibile se e solo se $x \in L$.

Poiché $L \in \text{NP}$, esiste una macchina di Turing non deterministica M che decide L in tempo $p(n)$ per qualche polinomio p . Un calcolo accettante di M su x è una sequenza di configurazioni $C_0, C_1, \dots, C_{p(n)}$ dove C_0 è la configurazione iniziale, $C_{p(n)}$ è una configurazione accettante, e ogni C_{i+1} è ottenibile da C_i mediante una transizione valida di M .

La formula $\varphi_{\{L,x\}}$ codifica questa struttura: introduce variabili proposizionali per ogni tripla (cella del nastro, posizione temporale, simbolo), per ogni coppia (stato, posizione temporale), e aggiunge clausole che rappresentano: (i) la configurazione iniziale di M su x ; (ii) le regole di transizione di M — per ogni tripla (stato, simbolo, passo temporale), le clausole che specificano lo stato successivo e il simbolo scritto; (iii) la condizione di accettazione — almeno uno degli stati accettanti deve essere raggiunto entro $p(n)$ passi.

Il numero di variabili è $O(p(n)^2)$ — polinomiale in n — e il numero di clausole è $O(p(n)^2)$ — polinomiale in n . La costruzione è eseguibile in tempo polinomiale. La correttezza della riduzione segue per costruzione: una macchina accettante corrisponde a un'assegnazione soddisfacente, e viceversa.

La bellezza della dimostrazione di Cook è che ripete la stessa mossa di Gödel: codifica il comportamento di un sistema computazionale (la macchina di Turing) in un sistema simbolico (la formula proposizionale),

sfruttando il fatto che la computazione ha struttura finita e locale che si lascia catturare da simboli. Gödel codificava la sintassi della dimostrazione nell'aritmetica; Cook codifica la computazione della macchina nella logica proposizionale. In entrambi i casi, la codifica rivela che il sistema ha potere espressivo sufficiente a parlare di se stesso — e questa auto-descrizione produce risultati di impossibilità o di completezza universale.

E.6 La fenomenologia della scoperta matematica: Poincaré, Hadamard e l'inconscio

Prima di connettere la complessità computazionale alla creatività matematica in modo più diretto, è necessario considerare la testimonianza introspettiva dei matematici sulla propria esperienza di scoperta. Questa testimonianza è epistemicamente fragile — l'introspezione non è un metodo affidabile per indagare i meccanismi cognitivi — ma è l'unica evidenza diretta disponibile e ha una coerenza interna che merita attenzione.

Henri Poincaré, nella conferenza *L'invention mathématique* del 1908 poi raccolta in *Science et Méthode* (1908), descrive con precisione insolita il processo che portò alla scoperta delle funzioni fuchsiane. Poincaré racconta di aver lavorato intensamente al problema per quindici giorni senza progressi, poi di averlo abbandonato per un viaggio. Al momento di salire su un autobus a Coutances, senza aver pensato al problema, la dimostrazione si presentò alla sua mente completa: le funzioni fuchsiane erano identiche alle trasformazioni della geometria non-euclidea. Poincaré tornò a Caen, si sedette, e verificò la risultato in poche ore.

Jacques Hadamard, nel volume *An Essay on the Psychology of Invention in the Mathematical Field* del 1945 — uno dei testi più seri sulla psicologia della

creatività matematica — sistematizzò le testimonianze di Poincaré e di altri matematici (inclusa una corrispondenza con Einstein) in un modello in quattro fasi: preparazione (studio intenso del problema), incubazione (abbandono consapevole), illuminazione (la soluzione emerge spontaneamente), verifica (controllo formale della soluzione). Hadamard notò che quasi universalmente, tra i matematici che aveva intervistato, la fase di illuminazione avveniva in momenti di rilassamento — passeggiate, risvegli, momenti prima del sonno — mai durante il lavoro concentrato.

Hadamard propose che la scoperta matematica coinvolgesse processi inconsci — al di sotto del livello della coscienza verbale — che operavano in parallelo al pensiero consapevole e producevano combinazioni di idee che il pensiero consapevole non aveva raggiunto deliberatamente. Questa proposta era speculativa nel 1945, ma trova un supporto formale sorprendente nella teoria delle reti neurali: i processi di attivazione distribuita che avvengono nelle reti neurali profonde sono precisamente processi "inconsci" nel senso che operano in parallelo, senza sequenzialità e senza simbolizzazione esplicita, e producono output che non sono direttamente tracciabili al processo che li ha generati.

E.7 La complessità dell'intuizione come compressione

La connessione tra la fenomenologia di Poincaré-Hadamard e la teoria della complessità algoritmica può essere formulata in termini precisi attraverso il concetto di compressione. L'intuizione matematica — il salto che porta direttamente alla soluzione senza esplorazione sistematica dello spazio delle possibilità — è funzionalmente equivalente a una **compressione dello spazio di ricerca**: invece di esplorare un esponenziale di possibilità, l'intuizione seleziona direttamente (o quasi direttamente) la direzione corretta.

Nella terminologia della complessità algoritmica, un'intuizione è un algoritmo euristico che, su una distribuzione di problemi con struttura specifica, produce soluzioni in tempo molto inferiore al worst-case. Formalmente, sia L un problema NP-completo e sia D una distribuzione sui suoi istanze. Un algoritmo euristico A è **efficiente rispetto a D** se il tempo atteso di A sull'insieme di istanze estratte da D è polinomiale, anche se il worst-case su tutte le istanze è esponenziale. La congettura $P \neq NP$ riguarda il worst-case, non il caso medio sulle distribuzioni naturali.

Questo è esattamente il meccanismo dei sistemi neurali addestrati: apprendono struttura dalla distribuzione dei problemi incontrati durante l'addestramento, e questa struttura consente loro di risolvere efficientemente i problemi della distribuzione anche quando il problema generale è NP-difficile. Il cervello del matematico esperto fa la stessa cosa: anni di esposizione a dimostrazioni matematiche addestrano le reti neurali del cervello a riconoscere le strutture tipiche delle dimostrazioni, e questa struttura appresa consente l'intuizione — la navigazione efficiente dello spazio delle possibilità grazie a un prior ben calibrato sulle soluzioni probabili.

L'illuminazione di Poincaré sull'autobus non è misteriosa in questa prospettiva: è il momento in cui i processi neurali distribuiti, attivati dall'esposizione prolungata al problema durante la fase di preparazione, convergono verso una configurazione di bassa energia — un minimo nell'energia libera friston-iana — che corrisponde alla soluzione. Il rilassamento durante il viaggio in autobus riduce l'interferenza dei processi cognitivi consapevoli e sequenziali, permettendo ai processi distribuiti e paralleli di convergere liberamente. La soluzione emerge non nonostante l'assenza di pensiero deliberato, ma grazie ad essa.

E.8 Il teorema di Baker-Gill-Solovay e i limiti delle tecniche note

Un aspetto della questione P vs NP che illustra la profondità del problema — e la sua connessione con i teoremi di incompletezza — è il **teorema di relativizzazione** di Baker, Gill e Solovay (1975). Questo teorema mostra che le tecniche dimostrative standard non possono risolvere P vs NP .

Un **oracolo** è una funzione aggiuntiva $O: \{0,1\}^* \rightarrow \{0,1\}$ che una macchina di Turing può interrogare in un singolo passo — come una funzione di libreria esterna. P^O e NP^O sono le classi P e NP relative all'oracolo O : le classi di problemi risolvibili rispettivamente in tempo polinomiale deterministico e non-deterministico con accesso a O .

Il teorema di Baker-Gill-Solovay afferma:

(i) Esiste un oracolo A tale che $P^A = NP^A$. (ii) Esiste un oracolo B tale che $P^B \neq NP^B$.

Questo significa che la questione P vs NP non è *relativizzabile*: non c'è risposta uniforme valida per tutti gli oracoli. Le tecniche dimostrative che "relativizzano" — che producono lo stesso risultato per qualunque oracolo — non possono decidere P vs NP , perché per certi oracoli la risposta è sì e per altri no.

La conseguenza è che qualunque dimostrazione di $P \neq NP$ deve impiegare proprietà specifiche della computazione senza oracoli — proprietà che si rompono in presenza di oracoli arbitrari. Le tecniche diagonali standard, che relativizzano, sono escluse. Si richiede una comprensione più profonda della struttura aritmetica e combinatoria della computazione.

Questo ha un'eco con il secondo teorema di incompletezza di Gödel: così come un sistema non può dimostrare la propria coerenza dall'interno, le tecniche standard di dimostrazione non possono risolvere P vs NP dall'interno del loro stesso paradigma. Entrambi i problemi richiedono un salto a un livello meta-teorico più alto — e il salto richiesto per P vs NP è ancora più oscuro di quello che Gödel aveva già reso visibile nel dominio della logica.

E.9 Algoritmi e creatività: l'ipotesi di Levin

Leonid Levin, co-scopritore indipendente dell'NP-completezza, ha proposto un algoritmo di ricerca universale — il **tempo ottimale universale** — che ha conseguenze filosofiche notevoli per la questione della creatività algoritmica. L'algoritmo di Levin (1973), detto Lsearch, è il seguente:

Per trovare una soluzione a un problema specificato da un predicato verificabile $V(x, s)$ — "s è una soluzione per il problema x" — Lsearch esegue in parallelo tutti i programmi $p_1, p_2, p_3, \dots$ nell'ordine della loro lunghezza crescente, dedicando al programma p_i una frazione $2^{-|p_i|}$ del tempo totale. Quando un programma p_i produce una stringa s tale che $V(x, s) = 1$, Lsearch si ferma e restituisce s.

Il risultato di Levin è: se il problema ha una soluzione, Lsearch la trova in tempo al più polinomiale nel tempo del programma più corto che la trova, moltiplicato per una costante che dipende solo dalla lunghezza di quel programma — non dalla dimensione del problema. Formalmente, se esiste un programma p^* di lunghezza l^* che risolve il problema in tempo T, allora Lsearch risolve il problema in tempo $O(2^{\{l^*\}} \cdot T)$.

L'algoritmo di Levin è ottimale nel senso che nessun altro algoritmo di ricerca può essere sistematicamente migliore di un fattore

superpolinomiale: Lsearch trova la soluzione al più con overhead moltiplicativo costante ($2^{\{l^*\}}$) rispetto al miglior programma che la trova. Esso realizza il prior universale di Solomonoff applicato alla ricerca: simula tutti i programmi possibili con probabilità proporzionale alla loro semplicità, e trova la soluzione non appena il programma più semplice che la produce ha avuto abbastanza tempo.

La conseguenza filosofica: se $P \neq NP$, l'algoritmo di Levin non è polinomiale nel worst case — il fattore $2^{\{l^*\}}$ può essere esponenziale. Ma mostra che la creatività algoritmica — trovare nuove soluzioni — è in linea di principio catturabile da un programma universale che simula tutti i programmi possibili. La creatività non è misteriosa: è la capacità di trovare il programma più corto che risolve il problema. E questa capacità è essa stessa computabile — solo non polinomiale nel caso peggiore.

Questo è l'erede algoritmico della posizione di Laplace: l'universo è deterministico e, con abbastanza risorse, calcolabile. La creatività matematica è computabile — Lsearch è una prova di esistenza — ma richiede risorse che nel caso pessimo sono esponenziali. L'intuizione del matematico esperto è l'euristica che naviga verso il programma corto sfruttando la struttura della distribuzione dei problemi matematici, comprimendo il fattore esponenziale $2^{\{l^*\}}$ in qualcosa di molto più gestibile.

E.10 Implicazioni per la tesi centrale

Le considerazioni di questa espansione aggiungono una dimensione alla tesi centrale della monografia che le parti precedenti non avevano sviluppato. L'asimmetria verifica-scoperta catturata dalla congettura $P \neq NP$ è la manifestazione computazionale dell'asimmetria fondamentale tra

usare la logica come strumento di verifica e come meccanismo di scoperta.

La logica formale è, essenzialmente, uno strumento di verifica: data una sequenza di passi proposta come dimostrazione, la logica verifica se ogni passo è corretto. Questa verifica è in P (per sistemi logici standard): un proof checker è un programma efficiente. Ma la *ricerca* di una dimostrazione — la scoperta del percorso da assiomi a teorema — è un problema NP-difficile: SAT è riducibile alla ricerca di dimostrazioni proposizionali, e i theorem provers automatici scalano esponenzialmente con la profondità delle dimostrazioni necessarie.

I sistemi cognitivi intelligenti — biologici e artificiali — non verificano: scoprono. Navigano lo spazio delle possibilità sfruttando la struttura statistica della distribuzione dei problemi, producendo soluzioni che in seguito si lasciano verificare formalmente. La logica interviene *dopo* la scoperta per certificarla, non prima per produrla. Questo è esattamente il modello di Poincaré: l'intuizione trova la soluzione, la verifica formale la certifica.

E questo è esattamente il modo in cui l'AI neurale opera: produce output che si lasciano valutare come corretti o scorretti (verifica), ma il meccanismo di produzione non è la ricerca sistematica nello spazio delle possibilità — è la navigazione statistica in uno spazio di alta dimensione guidata dalla struttura appresa dalla distribuzione di addestramento. La logica è lo strumento di verifica. L'apprendimento statistico è il meccanismo di scoperta. I due sono complementari, non concorrenti.

Espansione — Parte F

La struttura temporale dell'intelligenza, l'atemporalità della logica, e il problema della coscienza

F.1 Un'asimmetria trascurata: il tempo

C'è un'asimmetria tra la logica formale e i sistemi cognitivi reali — biologici e artificiali — che la monografia ha lasciato implicita pur essendo pervasiva: la **struttura temporale**. La logica classica è atemporale. Le proposizioni logiche non hanno indice temporale: $A \wedge B$ è vera o falsa, non vera-alle-ore-14:30-del-3-marzo. Le dimostrazioni sono oggetti astratti, sequenze di formule in un ordine logico che non è un ordine temporale. Il modello semantico — l'interpretazione I in un dominio D — è statico: assegna valori di verità a proposizioni senza riferimento al tempo.

I sistemi cognitivi reali, al contrario, sono costitutivamente temporali. Il cervello elabora l'informazione in sequenze di millisecondi: un potenziale d'azione dura circa 1 ms, un'oscillazione gamma corticale (30–90 Hz) dura 11–33 ms, un ciclo theta ippocampale (4–8 Hz) dura 125–250 ms, un ciclo alfa (8–12 Hz) dura 83–125 ms. Il ragionamento consapevole ha tempi dell'ordine dei secondi. Il ciclo sonno-veglia è di 24 ore. L'apprendimento a lungo termine ha scale temporali di giorni, mesi, anni. L'intelligenza biologica non è un sistema che produce risposte a stimoli: è un sistema che si svolge nel tempo, con strutture a scale temporali multiple e annidate.

I modelli linguistici transformer hanno una relazione particolare con il tempo: elaborano sequenze, ma lo fanno in modo parallelo — l'attenzione calcola le relazioni tra tutte le posizioni della sequenza simultaneamente. Questo parallelo processing è la fonte dell'efficienza computazionale dei transformer rispetto alle architetture ricorrenti, ma

produce una rappresentazione del tempo radicalmente diversa da quella del cervello biologico: il passato e il futuro della sequenza sono equidistanti nel calcolo dell'attenzione, ponderati dai punteggi di attenzione senza asimmetria intrinseca.

F.2 La logica temporale e i suoi limiti

La logica non è rimasta indifferente al problema del tempo: la **logica temporale** — sviluppata da Prior negli anni Cinquanta e poi formalizzata in versioni diverse da Pnueli, Clarke, Emerson e Sistla per la verifica dei sistemi reattivi — estende la logica proposizionale con operatori modali temporali. I principali operatori sono:

$G\varphi$: φ è vera in tutti gli istanti futuri (*globally*, o *always*) $F\varphi$: φ è vera in qualche istante futuro (*finally*, o *eventually*) $X\varphi$: φ è vera nell'istante immediatamente successivo (*next*) $\varphi \text{ U } \psi$: φ è vera fino a quando ψ diventa vera (*until*)

e i corrispondenti passati: $H\varphi$ (sempre nel passato), $P\varphi$ (in qualche momento nel passato), $Y\varphi$ (nell'istante precedente).

La logica temporale lineare (LTL) interpreta queste formule su sequenze lineari di istanti. La logica ad albero computazionale (CTL e CTL*) le interpreta su strutture ad albero dove dal presente si ramificano futuri alternativi, permettendo di quantificare su percorsi: $AG\varphi$ (su ogni percorso, φ è sempre vera), $EF\varphi$ (esiste un percorso su cui φ diventa eventualmente vera).

La verifica di sistemi reattivi — software di controllo, protocolli di comunicazione, sistemi embedded — attraverso la model checking (Clarke e Emerson 1981, Quille e Sifakis 1982) è una delle applicazioni industriali più solide della logica formale: dato un modello finito-stato di un sistema e una specifica in logica temporale, la model checking

determina automaticamente se il sistema soddisfa la specifica. Il modello checking ha rilevato bug profondi in hardware e software che sfuggivano al testing tradizionale, ed è oggi una componente standard dello sviluppo di processori e protocolli sicuri.

Ma la logica temporale mantiene l'assunzione fondamentale della logica classica: le proposizioni hanno valori di verità discreti (vere o false), le transizioni tra stati sono istantanee e atomiche, il tempo è discreto e lineare o ramificato. Questa struttura non cattura la continuità, il degrado graduale, il rumore, l'ambiguità — proprietà fondamentali dei sistemi cognitivi reali. La logica temporale è potente per i sistemi digitali chiusi; non è adeguata per la cognizione biologica aperta.

F.3 La struttura oscillatoria del cervello come elaborazione temporale

L'elaborazione temporale nel cervello biologico non è sequenziale nel senso discreto della logica temporale: è **oscillatoria**. Il cervello mantiene sincronizzazioni di fase tra oscillazioni a diverse frequenze in diverse regioni corticali, e questa struttura oscillatoria è il meccanismo attraverso cui diverse aree cerebrali coordinano la propria elaborazione.

Il ruolo delle oscillazioni gamma (30–90 Hz) nell'associazione di caratteristiche (binding problem) è stato proposto da Wolf Singer e Francis Crick: l'ipotesi è che le popolazioni di neuroni che codificano caratteristiche dello stesso oggetto percettivo sincronizzino le proprie oscillazioni gamma, creando una coalizione temporanea che costituisce la rappresentazione dell'oggetto come unità. Questa ipotesi del binding-by-synchrony è stata empiricamente controversa — la sincronizzazione gamma si osserva, ma la sua causalità per il binding non

è stabilita — ma ha stimolato una ricerca estesa sulla funzione computazionale delle oscillazioni cerebrali.

Le oscillazioni theta ippocampali (4–8 Hz) hanno un ruolo più documentato: modulano la codifica e il recupero delle memorie episodiche. La scoperta dei "place cells" di O'Keefe (1971) e dei "grid cells" di Moser e Moser (2005) — neuroni nell'ippocampo e nella corteccia entorinale che si attivano in posizioni specifiche dello spazio fisico — ha mostrato che l'ippocampo implementa una mappa spaziale dell'ambiente. Le oscillazioni theta modulano la fase di attivazione di queste cellule in modo da codificare la sequenza di posizioni attraversate, permettendo la "replay" delle sequenze durante il sonno e il consolidamento della memoria spaziale ed episodica nella neocorteccia.

Il ciclo theta è fondamentale per una ragione computazionale precisa: implementa una forma di **compressione temporale** delle esperienze sequenziali. Un percorso che richiede minuti per essere esplorato fisicamente viene codificato come sequenza di attivazioni nell'arco di pochi cicli theta (centinaia di millisecondi), permettendo alla plasticità sinaptica Hebbian di rafforzare le connessioni tra neuroni co-attivi in sequenza rapida. Il theta è il meccanismo biologico che accelera l'apprendimento per associazione.

F.4 Il problema del legame e la logica della percezione

Il **binding problem** — come il cervello integra le caratteristiche percettive di un oggetto elaborate in aree diverse in una percezione unificata — è uno dei problemi più profondi delle neuroscienze cognitive, e ha implicazioni dirette per la questione di questa monografia. Un oggetto visivo — un pomodoro rosso, rotondo, lucente, di odore caratteristico — attiva contemporaneamente neuroni nel V4 (colore),

V5/MT (movimento), aree temporali (forma), corteccia olfattiva (odore), e molte altre regioni. Come vengono "legati" questi segnali in un'unica esperienza percettiva coerente?

Il problema del legame non ha risposta nella logica classica: la logica può rappresentare il fatto che l'oggetto ha le proprietà in questione — $\exists x [\text{Rosso}(x) \wedge \text{Rotondo}(x) \wedge \text{Lucente}(x)]$ — ma la rappresentazione proposizionale non cattura l'*integrazione* delle caratteristiche nell'esperienza. Nessun assioma o regola inferenziale produce l'unità della percezione a partire dalla molteplicità delle caratteristiche.

La risposta computazionale che ha più trazione empirica è la sincronizzazione oscillatoria, ma questa risposta è essenzialmente temporale, non logica: l'unità percettiva non è una proprietà logica dell'oggetto ma una proprietà dinamica della rete neurale che lo elabora — la sincronizzazione delle oscillazioni tra le aree che codificano le sue caratteristiche. L'integrazione è un evento temporale, non una deduzione.

Questo illustra un principio generale: molte delle funzioni cognitive che sembrano richiedere rappresentazioni simboliche — la coerenza della percezione, l'unità del pensiero, la continuità dell'identità personale — sono in realtà prodotte da meccanismi dinamici temporali che non hanno struttura proposizionale. La logica classica è lo strumento sbagliato per descrivere questi meccanismi non perché sia sbagliata nel proprio dominio, ma perché il suo dominio esclude essenzialmente la dimensione temporale.

F.5 L'informazione integrata e la misura della coscienza

Il problema della coscienza — il *hard problem* di Chalmers: perché c'è esperienza soggettiva, perché i processi fisici sono accompagnati da qualia — rimane il più ostinato della filosofia della mente. Non è mia

intenzione proporre una soluzione a ciò che nessuno ha risolto. Ma la teoria dell'**Informazione Integrata** (IIT) di Giulio Tononi ha abbastanza struttura formale precisa da meritare un'analisi critica nel contesto di questa monografia, e offre un caso interessante in cui un tentativo di quantificare la coscienza rivela le proprie connessioni e disconnessioni con i temi qui sviluppati.

La IIT è fondata sull'intuizione che la coscienza sia identica all'informazione integrata — l'informazione che il sistema possiede come insieme, al di là di quella posseduta dalle sue parti. La misura di questa informazione integrata è la quantità φ (phi), definita operativamente come la differenza tra l'informazione generata dal sistema nel suo insieme e la somma dell'informazione generata dalle sue parti minimalmente interconnesse.

La definizione formale nella versione IIT 3.0 è la seguente. Si consideri un sistema S di n elementi in uno stato s . Per ogni possibile partizione bipartita (A, B) del sistema — ogni modo di suddividere S in due sottoinsiemi — si misura l'**earth mover's distance** (o Wasserstein distance) tra la distribuzione causale del sistema intero e quella del sistema partizionato:

$$\varphi(M, Z) = W_1(P(S_{t-1} \mid S_t = s), P_{\text{cut}}(S_{t-1} \mid S_t = s))$$

dove P è la distribuzione causale del sistema (le probabilità degli stati passati dato lo stato attuale) e P_{cut} è la stessa distribuzione con la partizione M (con le connessioni tra A e B tagliate). La misura di informazione integrata del sistema è:

$$\Phi = \min_M \varphi(M, S)$$

il minimo su tutte le partizioni bipartite del sistema. Se $\Phi = 0$, il sistema non ha coscienza: la sua informazione è completamente decomponibile nelle parti. Se $\Phi > 0$, il sistema è cosciente in misura proporzionale a Φ .

La IIT ha proprietà formali interessanti. Prevede che il cervello abbia Φ molto alto — a causa della sua enorme connettività integrata — mentre un computer classico con la stessa funzione input-output abbia Φ molto basso, perché la computazione sequenziale è decomponibile in passi indipendenti. Questo è un argomento teorico, non empirico, per distinguere la coscienza biologica dalla coscienza artificiale: i sistemi che eseguono computazione sequenziale, per quanta competenza cognitiva esibiscano, avrebbero Φ vicino a zero.

F.6 Le difficoltà della IIT

La IIT soffre di difficoltà gravi su cui occorre essere onesti. La prima è computazionale: calcolare Φ esattamente richiede considerare tutte le $2^n - 2$ partizioni bipartite di un sistema di n elementi, che cresce esponenzialmente con n . Per un sistema con $n = 100$ neuroni, le partizioni sono $2^{\{100\}} - 2 \approx 10^{\{30\}}$: il calcolo esatto è computazionalmente intrattabile per qualunque sistema di dimensione biologica. Questo non è solo un problema pratico: significa che Φ non è una misura operazionalmente realizzabile per il cervello, e qualunque stima è necessariamente un'approssimazione grossolana di un ideale non calcolabile.

La seconda difficoltà è empirica: la IIT prevede che qualunque sistema con alta connettività e integrazione abbia Φ alto, indipendentemente dalla sua architettura fisica. Questo include potenzialmente le reti neurali artificiali deep addestrate con architetture ricorrenti — che hanno alto grado di connettività — e persino certi sistemi fisici semplici ma

altamente integrati. La previsione che certi reticoli cristallini possano avere Φ non nullo è stata criticata come controintuitiva: nessuno ritiene che un cristallo di silicio abbia esperienza soggettiva.

La terza difficoltà è l'incompatibilità con il funzionalismo: la IIT afferma che due sistemi con la stessa funzione input-output ma diversa struttura interna possono avere valori di Φ radicalmente diversi. Questo implica che la realizzabilità multipla di Putnam è falsa per la coscienza: la coscienza non è una proprietà funzionale ma una proprietà strutturale-intrinseca. Questo è un impegno ontologico controverso che la IIT assume senza argomentarlo in modo indipendente.

Nonostante queste difficoltà, la IIT ha un merito che va riconosciuto: è uno dei pochi tentativi seri di dare una definizione formale di coscienza che produca previsioni empiriche testabili. La previsione che la sindrome di locked-in — in cui i pazienti sono coscienti ma paralizzati — abbia Φ alto mentre il sonno profondo abbia Φ basso è stata parzialmente confermata da studi con la TMS (stimolazione magnetica transcranica) combinata con EEG: la complessità della risposta corticale alla TMS — una misura empirica approssimativa di Φ — è alta nei soggetti svegli e nei pazienti locked-in, bassa nel sonno profondo e nell'anestesia.

F.7 Il principio di causalità intrinseca e la sua connessione con l'energia libera

La IIT e il principio dell'energia libera di Friston, pur sviluppandosi in modo indipendente, hanno una connessione strutturale che merita esplicitazione. Entrambi cercano di caratterizzare i sistemi cognitivi attraverso una misura scalare che cattura la qualità dell'integrazione informazionale del sistema.

Nella IIT, la misura è Φ : la quantità di informazione causale che il sistema genera come insieme irriducibile alle sue parti. Nella teoria di Friston, la misura è l'energia libera F : la divergenza tra il modello generativo interno e la distribuzione delle cause degli input sensoriali. Entrambe le misure sono funzioni della struttura causale del sistema e della sua distribuzione di stati.

La connessione formale può essere articolata nel seguente modo. L'energia libera variazionale è:

$$F = D_{\text{KL}}(Q(s) \parallel P(s|o)) - \log P(o)$$

dove D_{KL} misura la divergenza tra la distribuzione interna Q e la posteriore reale $P(s|o)$. Un sistema che minimizza F approssima la distribuzione causale reale del mondo. L'informazione integrata Φ misura quanto la distribuzione causale del sistema dipenda dall'interazione delle sue parti — quanto sia irriducibile. Un sistema con Φ alto ha un modello causale interno che non è decomponibile in parti indipendenti: le sue componenti informano mutuamente le loro previsioni, producendo un modello del mondo più ricco di quanto le parti singole potrebbero produrre.

L'ipotesi speculativa — ma suggestiva — è che i sistemi con Φ alto siano sistemi che minimizzano l'energia libera in modo più efficiente: l'integrazione causale tra le parti permette al sistema di costruire un modello generativo del mondo che cattura dipendenze statistiche di ordine superiore che un sistema decomponibile non potrebbe catturare. La coscienza — se la IIT è corretta nell'identificarla con Φ — sarebbe la proprietà che emerge quando l'integrazione causale raggiunge il livello necessario per costruire modelli generativi del mondo di sufficiente ricchezza.

Questa speculazione non è confermata empiricamente, ma ha la struttura di un'ipotesi scientifica testabile: predice che i sistemi con Φ alto dovrebbero avere prestazioni superiori nei compiti che richiedono cattura di dipendenze di ordine superiore, e che questa superiorità dovrebbe scomparire quando i sistemi vengono partizionati artificialmente in componenti meno integrate.

F.8 Il tempo, la logica e la cognizione incarnata

La dimensione temporale dell'intelligenza ha una rilevanza specifica per la questione del rapporto tra logica e cognizione che questa monografia ha sviluppato. La logica classica, come si è osservato, è atemporale: i suoi teoremi sono veri per sempre o non sono veri, le sue dimostrazioni sono valide indipendentemente dal momento in cui vengono eseguite. Questa atemporalità è una virtù per i problemi che la logica serve meglio — la matematica pura, la verifica formale — ma è una limitazione strutturale per modellare la cognizione reale.

La cognizione incarnata — una posizione sviluppata da Varela, Thompson e Rosch nel volume *The Embodied Mind* del 1991, con radici nella fenomenologia di Merleau-Ponty — argomenta che la cognizione non è elaborazione di rappresentazioni astratte ma enazione: il dispiegarsi di un'interazione continua e temporale tra un organismo vivente e il suo ambiente. La cognizione non avviene *in* un agente che poi *agisce sul* mondo: avviene *come* l'interazione dell'agente con il mondo, in una temporalità che non si lascia separare dalla struttura fisica del corpo.

In questa prospettiva, la logica formale è un'astrazione che rimuove deliberatamente la temporalità e l'incarnazione: opera su simboli privi di corpo, in un tempo che è ordine logico ma non durato. Questa rimozione

è cognitivamente costosa — richiede uno sforzo di astrazione che il cervello non esegue spontaneamente — ed è produttiva solo in domini dove le proprietà rimosse (temporalità, incarnazione, contesto) sono irrilevanti per il problema.

L'AI neurale moderna — i transformer, le reti ricorrenti, i modelli multimodali che integrano testo, immagine e suono — si avvicina alla cognizione incarnata più di quanto l'AI simbolica abbia mai fatto. Non perché abbia un corpo, ma perché la sua struttura temporale (le sequenze) e la sua capacità di integrare informazioni da modalità diverse (i modelli multimodali) riproducono aspetti della struttura dell'interazione organismo-ambiente che la logica atemporale ignora.

F.9 Verso una teoria unificata: le cinque dimensioni della cognizione

Le espansioni di questa monografia hanno aggiunto, alle cinque affermazioni finali dell'Espansione D, un insieme di dimensioni dell'intelligenza che la logica formale non cattura strutturalmente:

La **dimensione della complessità**: la cognizione opera nell'intersezione tra problemi computazionalmente difficili nel caso peggiore e distribuzioni di problemi reali con struttura che permette approssimazioni efficienti. La congettura $P \neq NP$ delimita lo spazio di ciò che è possibile; l'apprendimento statistico popola quel spazio con soluzioni praticamente efficaci.

La **dimensione temporale**: la cognizione si svolge in strutture temporali gerarchiche — dalle oscillazioni gamma alle memorie a lungo termine — che non hanno analogo nella logica atemporale. L'integrazione temporale dell'informazione, la sincronizzazione oscillatoria, il replay ippocampale: sono meccanismi cognitivi fondamentali che la logica non modella.

La **dimensione causale**: la cognizione è embedded in strutture causali — il modello generativo del cervello, l'informazione integrata della IIT — che hanno proprietà irriducibili alle loro componenti. La struttura causale integrata è la condizione per la produzione di modelli del mondo sufficientemente ricchi.

La **dimensione evolutiva**: la cognizione è il prodotto di una co-evoluzione gene-cultura che ha selezionato per meccanismi di apprendimento efficienti sulle distribuzioni di problemi degli ambienti ancestrali. Questi meccanismi sono statistici e probabilistici, non logici.

La **dimensione semantica**: la cognizione è radicata nel mondo attraverso modelli generativi costruiti dall'interazione con l'ambiente — il grounding come struttura del modello generativo — e questa struttura di radicamento è distribuita, dinamica e irriducibile alla designazione simbolica.

Nessuna di queste cinque dimensioni è catturata dalla logica formale. Tutte e cinque sono presenti, in misura variabile, nell'intelligenza biologica. L'intelligenza artificiale neurale cattura in modo approssimativo le dimensioni della complessità, quella evolutiva (attraverso l'addestramento) e quella semantica (attraverso il grounding distribuito); cattura solo parzialmente quella temporale (le sequenze, ma non le oscillazioni); e non cattura per nulla quella causale nel senso della IIT. Questo inventario non è un elenco di difetti da correggere: è una mappa della distanza tra i sistemi attuali e la comprensione piena di cosa sia l'intelligenza. Una mappa è utile precisamente quando non si è ancora arrivati a destinazione.

La logica formale occupa, in questa mappa, una posizione precisa: è lo strumento che cattura perfettamente la dimensione della verifica formale — la correttezza delle inferenze in sistemi chiusi — e non cattura

nessuna delle altre quattro dimensioni. La sua potenza è reale e insostituibile nel suo dominio. La sua pretesa di essere il fondamento dell'intelligenza è confutata dalla struttura di questa mappa.

Nota bibliografica dell'espansione E-F

Per P vs NP e la complessità computazionale: Cook, *The Complexity of Theorem-Proving Procedures*, in *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (1971); Karp, *Reducibility among Combinatorial Problems*, in *Complexity of Computer Computations*, a cura di R. Miller e J. Thatcher (1972); Levin, *Universal Sequential Search Problems*, in *Problems of Information Transmission*, 9 (1973); Baker, Gill e Solovay, *Relativizations of the $P = NP$ Question*, in *SIAM Journal on Computing*, 4 (1975); Garey e Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (1979).

Per la logica temporale e la model checking: Prior, *Time and Modality* (1957); Pnueli, *The Temporal Logic of Programs*, in *Proceedings of the 18th Annual Symposium on Foundations of Computer Science* (1977); Clarke ed Emerson, *Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic*, in *Logic of Programs*, Lecture Notes in Computer Science 131 (1981); Quielle e Sifakis, *Specification and Verification of Concurrent Systems in CESAR*, in *Proceedings of the 5th International Symposium on Programming* (1982).

Per la fenomenologia della scoperta matematica: Poincaré, *Science et Méthode* (1908); Hadamard, *An Essay on the Psychology of Invention in the Mathematical Field* (1945).

Per le oscillazioni cerebrali e il binding: Singer e Gray, *Visual Feature Integration and the Temporal Correlation Hypothesis*, in *Annual Review of Neuroscience*, 18 (1995); O'Keefe e Dostrovsky, *The Hippocampus as a Spatial*

Map, in *Brain Research*, 34 (1971); Moser, Kropff e Moser, *Place Cells, Grid Cells, and the Brain's Spatial Representation System*, in *Annual Review of Neuroscience*, 31 (2008); Buzsáki, *Rhythms of the Brain* (2006).

Per la teoria dell'informazione integrata: Tononi, *An Information Integration Theory of Consciousness*, in *BMC Neuroscience*, 5 (2004); Tononi, Boly, Massimini e Koch, *Integrated Information Theory: From Consciousness to Its Physical Substrate*, in *Nature Reviews Neuroscience*, 17 (2016); Massimini, Ferrarelli, Huber, Esser, Singh e Tononi, *Breakdown of Cortical Effective Connectivity during Sleep*, in *Science*, 309 (2005); Casali, Gosseries, Rosanova, Boly, Sarasso, Casali, Casarotto, Bruno, Laureys, Tononi e Massimini, *A Theoretically Based Index of Consciousness Independent of Sensory Processing and Behavior*, in *Science Translational Medicine*, 5 (2013).

Per la cognizione incarnata: Varela, Thompson e Rosch, *The Embodied Mind: Cognitive Science and Human Experience* (1991); Merleau-Ponty, *Phénoménologie de la perception* (1945); Clark, *Being There: Putting Brain, Body, and World Together Again* (1997); Chalmers, *Facing Up to the Problem of Consciousness*, in *Journal of Consciousness Studies*, 2 (1995).

Per la complessità media e l'aggirare $P \neq NP$: Levin, *Average Case Complete Problems*, in *SIAM Journal on Computing*, 15 (1986); Ben-David, Chor, Goldreich e Luby, *On the Theory of Average Case Complexity*, in *Journal of Computer and System Sciences*, 44 (1992); Bogdanov e Trevisan, *Average-Case Complexity*, in *Foundations and Trends in Theoretical Computer Science*, 2 (2006).

Appendice Tecnica I

L'aritmetizzazione completa della sintassi: funzioni primitive ricorsive, la predicabilità di Bew e la struttura formale della diagonalizzazione

I.1 Le funzioni primitive ricorsive: definizione formale

La classe delle funzioni **primitive ricorsive** (PR) è la più piccola classe di funzioni $\mathbb{N}^k \rightarrow \mathbb{N}$ (per $k \geq 0$) che contiene le funzioni base e che è chiusa rispetto alla composizione e alla ricorsione primitiva.

Le **funzioni base** sono tre:

- (i) La funzione zero: $Z: \mathbb{N} \rightarrow \mathbb{N}$, $Z(n) = 0$ per ogni n .
- (ii) Le funzioni di proiezione: per ogni $k \geq 1$ e ogni $1 \leq i \leq k$, $P_i^k: \mathbb{N}^k \rightarrow \mathbb{N}$, $P_i^k(n_1, \dots, n_k) = n_i$.
- (iii) La funzione successore: $S: \mathbb{N} \rightarrow \mathbb{N}$, $S(n) = n + 1$.

Le operazioni di chiusura sono due:

Composizione: se $g: \mathbb{N}^m \rightarrow \mathbb{N}$ e $h_1, \dots, h_k: \mathbb{N}^k \rightarrow \mathbb{N}$ sono PR, allora $f: \mathbb{N}^k \rightarrow \mathbb{N}$ definita da $f(x_1, \dots, x_k) = g(h_1(x_1, \dots, x_k), \dots, h_k(x_1, \dots, x_k))$ è PR.

Ricorsione primitiva: se $g: \mathbb{N}^k \rightarrow \mathbb{N}$ e $h: \mathbb{N}^{k+2} \rightarrow \mathbb{N}$ sono PR, allora $f: \mathbb{N}^{k+1} \rightarrow \mathbb{N}$ definita da:

- $f(0, x_1, \dots, x_k) = g(x_1, \dots, x_k)$
- $f(n+1, x_1, \dots, x_k) = h(n, f(n, x_1, \dots, x_k), x_1, \dots, x_k)$ è PR.

Dalla ricorsione primitiva si ottengono immediatamente le operazioni aritmetiche elementari. L'addizione: $\text{add}(0, y) = y = P_1^1(y)$; $\text{add}(n+1, y) = S(P_2^3(n, \text{add}(n, y), y)) = \text{add}(n, y) + 1$. La moltiplicazione: $\text{mult}(0, y) = Z(y) = 0$; $\text{mult}(n+1, y) = \text{add}(\text{mult}(n, y), y)$. La potenza: $\text{pow}(0, y) = S(Z(y)) = 1$; $\text{pow}(n+1, y) = \text{mult}(\text{pow}(n, y), y)$.

Crucialmente, le PR includono anche operazioni di manipolazione di sequenze finite attraverso la **codifica di Cantor** delle coppie: $\pi(m, n) = (m+n)(m+n+1)/2 + m$, una biiezione $\mathbb{N}^2 \rightarrow \mathbb{N}$ con funzioni di inversione π_1 e π_2 anch'esse PR. La codifica delle sequenze finite $\langle n_1, \dots, n_k \rangle$

avviene attraverso potenze di numeri primi, come nel corpo del testo principale.

I.2 La primitività ricorsiva delle relazioni sintattico-aritmetiche

Il nucleo tecnico dell'aritmetizzazione di Gödel è la dimostrazione che tutte le relazioni sintattiche tra espressioni formali diventano relazioni aritmetiche primitive ricorsive tra i loro numeri di Gödel. Definiamo il numero di Gödel di un'espressione attraverso la codifica a potenze di primi descritta nel corpo principale:

$$\ulcorner s_1 s_2 \dots s \urcorner = 2^{\{g(s_1)\}} \cdot 3^{\{g(s_2)\}} \cdot 5^{\{g(s_3)\}} \cdot \dots \cdot p_n^{\{g(s)\}}$$

dove g è l'assegnazione fissa di numeri naturali ai simboli primitivi del sistema e p è l' n -esimo numero primo. Definiamo le seguenti relazioni e funzioni e mostriamo che sono PR.

Lunghezza di una sequenza: $\text{Len}(x) =$ il numero di esponenti non-nulli nella fattorizzazione di x . Formalmente: $\text{Len}(x) = \mu y \leq x [p_{y+1} \nmid x]$ dove μ è il minimo limitato. La ricerca del minimo limitato — $\mu y \leq z [R(y)]$ — preserva la primitività ricorsiva se R è PR.

Estrazione del termine i -esimo: $(x)_i =$ l'esponente di p_i nella fattorizzazione di $x = \mu e \leq x [p_i^e \mid x]$. Questa funzione è PR.

Concatenazione: $\text{Cat}(x, y) =$ il numero di Gödel della sequenza ottenuta concatenando le sequenze codificate da x e da y . $\text{Cat}(x, y) = x \cdot \prod_{i=1}^{\text{Len}(y)} p_{\text{Len}(x)+i}^{\{(y)_i\}}$. Il prodotto è computato per ricorsione primitiva su $\text{Len}(y)$, che è limitato da y ; l'intera costruzione è PR.

Sostituzione: $\text{Sub}(x, v, t) =$ il numero di Gödel della formula ottenuta dalla formula con numero di Gödel x sostituendo tutte le occorrenze

libere della variabile con numero di Gödel v con il termine con numero di Gödel t . La costruzione di Sub richiede attraversare ricorsivamente la struttura della formula, identificare le occorrenze libere della variabile, e sostituirle. Ogni passo di questa procedura è PR, e la loro composizione è PR.

I.3 La primitività ricorsiva del predicato di dimostrazione

Il predicato più importante per il teorema di Gödel è $\text{Bew}(d, \varphi)$ — " d è il numero di Gödel di una dimostrazione della formula con numero di Gödel φ ". La sua PR richiede che siano PR:

(a) **$\text{Ax}(\mathbf{x})$** : " x è il numero di Gödel di un assioma". Per un sistema come PA, gli assiomi sono di forme schematiche (assiomi di Peano, schemi di assiomi logici) che si caratterizzano da proprietà strutturali delle espressioni. La chiave è che ogni schema assiomatico definisce una classe di formule ricorsivamente descrivibile: per esempio, lo schema di induzione afferma che per ogni formula $\varphi(v)$, la formula $\varphi(0) \wedge \forall v(\varphi(v) \rightarrow \varphi(Sv)) \rightarrow \forall v \varphi(v)$ è un assioma. Riconoscere se un'espressione è un'istanza di questo schema è una questione di analizzare la struttura dell'espressione — PR.

(b) **$\text{MP}(\mathbf{x}, \mathbf{y}, \mathbf{z})$** : " z si ottiene da x e y per modus ponens". Formalmente: $\text{MP}(x, y, z) \leftrightarrow (y \text{ è il numero di Gödel di una formula della forma } \varphi \rightarrow \psi) \wedge (x \text{ è il numero di Gödel di } \varphi) \wedge (z \text{ è il numero di Gödel di } \psi)$. Ogni condizione è una proprietà strutturale del numero di Gödel, riconoscibile in modo PR.

(c) **$\text{Proof}(\mathbf{d})$** : " d è il numero di Gödel di una sequenza di formule che costituisce una dimostrazione valida". Una dimostrazione è una sequenza $\langle \varphi_1, \dots, \varphi \rangle$ tale che ogni φ_i è o un assioma o si ottiene da due formule precedenti della sequenza per modus ponens. Formalmente: $\text{Proof}(d) \leftrightarrow$

$\forall i \leq \text{Len}(d) [\text{Ax}((d)_i) \vee \exists j < i \exists k < i \text{MP}((d), (d), (d)_i)]$. Il quantificatore universale limitato $\forall i \leq \text{Len}(d)$ preserva la PR; i quantificatori esistenziali limitati $\exists j < i$ e $\exists k < i$ analogamente. Quindi $\text{Proof}(d)$ è PR.

(d) **Bew(d, φ)**: " d è una dimostrazione della formula φ ". $\text{Bew}(d, \varphi) \leftrightarrow \text{Proof}(d) \wedge ((d)_{\text{Len}(d)} = \varphi)$, ossia d è una dimostrazione valida e la sua ultima formula è φ . Questo è PR.

(e) **Prov(φ)**: " φ è dimostrabile". $\text{Prov}(\varphi) \leftrightarrow \exists d \text{Bew}(d, \varphi)$. Questo è il passo critico: Prov è ricorsivamente enumerabile (Σ_1) ma non PR in generale — la quantificazione esistenziale illimitata non preserva la PR. Ma per ogni φ specifico, la funzione $\text{Prov}(\varphi)$ è calcolabile per enumerazione delle dimostrazioni.

La formula $\text{Prov}(y)$ del teorema di Gödel è una formula del linguaggio di S che esprime $\exists d \text{Bew}(d, y)$: la sua costruzione formale usa la PR di Bew e la quantificazione esistenziale nel linguaggio della logica del primo ordine. La rappresentabilità di Bew in S è garantita dal fatto che S è sufficientemente ricco da rappresentare tutte le relazioni PR: questo è il **Teorema di Rappresentabilità** che afferma che ogni funzione PR f è rappresentabile in PA nel senso che esiste una formula $F_f(x_1, \dots, x, y)$ tale che $\text{PA} \vdash F_f(n_1, \dots, n, m) \leftrightarrow f(n_1, \dots, n) = m$ per ogni scelta di valori $n_1, \dots, n, m$ (dove si usano i numerali $\bar{n}$ per i valori n).

I.4 Il lemma diagonale: forma generale e varianti

Il **lemma diagonale** (o lemma di punto fisso, o lemma di autoreferimento) è il dispositivo tecnico che permette di costruire formule che parlano di se stesse. Ne forniamo qui la forma generale e le varianti principali.

Forma standard: Per ogni formula $F(y)$ di S con una variabile libera y , esiste una sentenza G tale che:

$$S \vdash G \leftrightarrow F(\ulcorner G \urcorner)$$

Dimostrazione completa: Sia $D(y)$ la formula $F(\text{sub}(y, y))$, dove $\text{sub}(x, y)$ è la funzione primitiva ricorsiva tale che $\text{sub}(\ulcorner \varphi(v) \urcorner, n) = \ulcorner \varphi(\bar{n}) \urcorner$ — il numero di Gödel della formula ottenuta sostituendo la variabile libera di φ con il numerale $\bar{n}$. Questa funzione è PR come mostrato nella sezione precedente, dunque rappresentabile in S .

Sia $d = \ulcorner D(v) \urcorner$ il numero di Gödel della formula $D(y)$ con y come variabile libera. Definiamo $G = D(\bar{d}) = F(\text{sub}(\bar{d}, \bar{d}))$.

Verifichiamo che $S \vdash G \leftrightarrow F(\ulcorner G \urcorner)$. Calcoliamo $\text{sub}(\bar{d}, \bar{d})$: per definizione di sub , $\text{sub}(\ulcorner D(v) \urcorner, \bar{d}) = \ulcorner D(\bar{d}) \urcorner = \ulcorner G \urcorner$. Quindi $F(\text{sub}(\bar{d}, \bar{d}))$ è esattamente $F(\ulcorner G \urcorner)$. Ma G è definita come $F(\text{sub}(\bar{d}, \bar{d}))$, quindi $G = F(\ulcorner G \urcorner)$ come formule del linguaggio di S . Poiché S può dimostrare le identità tra numerali e i valori delle funzioni PR sui numerali, $S \vdash \text{sub}(\bar{d}, \bar{d}) = \ulcorner G \urcorner$, e dunque $S \vdash G \leftrightarrow F(\ulcorner G \urcorner)$. ■

Varianti del lemma diagonale:

(i) **Autoreferimento forte:** per ogni formula $F(y, z)$ con due variabili libere, esiste $G(z)$ con una variabile libera tale che $S \vdash G(z) \leftrightarrow F(\ulcorner G(z) \urcorner, z)$. Questa variante permette di costruire formule che si riferiscono a se stesse mentre mantengono un parametro libero.

(ii) **Punto fisso per predicati:** la versione delle logiche modali del lemma diagonale, usata dalla logica della dimostrabilità (GL di Löb), afferma che in GL ogni formula $A(p)$ ha un punto fisso modalemente equivalente: una sentenza B tale che $GL \vdash B \leftrightarrow A(B)$. La logica GL — con gli assiomi K (distribuzione della necessità), 4 ($\Box A \rightarrow \Box \Box A$), e il principio di Löb $\Box(\Box A \rightarrow A) \rightarrow \Box A$ — cattura esattamente la logica del predicato di dimostrabilità di PA.

I.5 Il teorema di Löb e la formalizzazione della dimostrazione gödeliana

Il **teorema di Löb** (1955) è il risultato che meglio illumina la struttura logica dei teoremi di Gödel. Esso afferma:

$$PA \vdash \Box A \rightarrow A \text{ implica } PA \vdash A$$

dove $\Box A$ abbrevia $\text{Prov}(\ulcorner A \urcorner)$. In parole: se PA può dimostrare "se A è dimostrabile allora A è vera", allora PA può dimostrare A direttamente.

La dimostrazione usa le **condizioni di derivabilità di Hilbert-Bernays-Löb** (HBL), tre proprietà del predicato di dimostrabilità che sono dimostrabili in PA:

(D1) $PA \vdash A$ implica $PA \vdash \Box A$ (se A è dimostrabile, allora è dimostrabile che A è dimostrabile)

(D2) $PA \vdash \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$ (la dimostrabilità di un'implicazione e del suo antecedente implica la dimostrabilità del conseguente)

(D3) $PA \vdash \Box A \rightarrow \Box \Box A$ (se A è dimostrabile, è dimostrabile che A è dimostrabile — la predicabilità è "introspettiva")

La dimostrazione del teorema di Löb procede come segue. Supponiamo $PA \vdash \Box A \rightarrow A$. Per il lemma diagonale, sia W una sentenza tale che $PA \vdash W \leftrightarrow (\Box W \rightarrow A)$ — W afferma "se sono dimostrabile allora A".

Dalla definizione di W: $PA \vdash W \rightarrow (\Box W \rightarrow A)$. Per D1: $PA \vdash \Box W \rightarrow \Box(\Box W \rightarrow A)$. (*)

Per D2: $PA \vdash \Box(\Box W \rightarrow A) \rightarrow (\Box \Box W \rightarrow \Box A)$. (**)

Per D3: $PA \vdash \Box W \rightarrow \Box \Box W$. (***)

Combinando (*), (**), (**): $PA \vdash \Box W \rightarrow \Box A$.

Per ipotesi, $PA \vdash \Box A \rightarrow A$. Dunque per transitività: $PA \vdash \Box W \rightarrow A$.

Ma W è definita come $\Box W \rightarrow A$, quindi: $PA \vdash W$. Per D1: $PA \vdash \Box W$.

Da $\Box W$ e $\Box W \rightarrow A$: $PA \vdash A$. ■

Il teorema di Löb mostra che la struttura della dimostrazione di Gödel non è un trucco ad hoc ma la manifestazione di proprietà profonde del predicato di dimostrabilità. Il secondo teorema di Gödel è un caso particolare: prendendo $A = \perp$ (la contraddizione), $\Box A \rightarrow A$ equivale a $\text{Prov}(\perp) \rightarrow \perp$, ossia $\neg \text{Prov}(\perp)$, ossia $\text{Con}(PA)$. Il teorema di Löb afferma allora: se $PA \vdash \text{Con}(PA)$, allora $PA \vdash \perp$ — ossia PA è incoerente. Assumendo la coerenza, $PA \not\vdash \text{Con}(PA)$.

I.6 La gerarchia aritmetica: struttura e completezza

La gerarchia aritmetica classifica gli insiemi di numeri naturali secondo la complessità della formula aritmetica che li definisce, misurata dal numero e dall'alternanza dei quantificatori.

Un insieme $A \subseteq \mathbb{N}$ è $\Sigma_0 = \Pi_0 = \Delta_0$ se è definibile da una formula con soli quantificatori limitati ($\forall x \leq t$ e $\exists x \leq t$ per qualche termine t). Questi insiemi sono tutti decidibili — la verifica è finita.

A è Σ_{+1} se è definibile da una formula $\exists x_1 \dots \exists x_n \varphi(x_1, \dots, x_n, y)$ dove φ è Π . A è Π_{+1} se è definibile da $\forall x_1 \dots \forall x_n \varphi(x_1, \dots, x_n, y)$ con $\varphi \Sigma$. A è Δ se è sia Σ che Π .

Le inclusioni sono strette: $\Delta \subsetneq \Sigma$, $\Delta \subsetneq \Pi$, e $\Sigma \cup \Pi \subsetneq \Delta_{+1}$. La prova della stretta inclusione usa diagonalizzazione: per ogni n , l'insieme $\{e : M_e \text{ termina su input } \ulcorner M_e \urcorner \text{ in } \leq n \text{ passi}\}$ è Σ -completo.

La collocazione degli insiemi fondamentali nella gerarchia:

Σ_1 : l'insieme dei teoremi di PA — $\{\ulcorner \varphi \urcorner : PA \vdash \varphi\}$. Questo è RE ma non decidibile (non Δ_1), per il secondo teorema di incompletezza.

Π_1 : il complemento dell'insieme dei teoremi, ma anche l'insieme dei teoremi universalmente veri nell'aritmetica standard — $\{\ulcorner \varphi \urcorner : \mathbb{N} \models \varphi \text{ per ogni sostituzione dei parametri liberi}\}$. Questo è Π_1 ma non Σ_1 .

Π_2 : l'insieme degli enunciati indecidibili — $\{\ulcorner \varphi \urcorner : PA \not\vdash \varphi \text{ e } PA \not\vdash \neg\varphi\}$. Non è difficile vedere che questo richiede quantificazione su dimostrazioni (Σ_1) della loro non-esistenza (Π_1), producendo Π_2 .

Δ_1 : gli insiemi decidibili. L'insieme dei numeri primi è Δ_1 ; il problema della fermata specifico per un programma fissato è Δ_1 (si esegue il programma e si aspetta).

La **gerarchia polinomiale** è la versione della gerarchia aritmetica nel regime della complessità computazionale: Σ^P e Π^P corrispondono a quantificatori alternanti esistenziali e universali su certificati verificabili in tempo polinomiale. $P = \Delta_1^P$, $NP = \Sigma_1^P$, $co-NP = \Pi_1^P$. La congettura $P \neq NP$ è l'ipotesi che la gerarchia polinomiale non collassi al primo livello.

I.7 La logica della dimostrabilità GL: completezza e teorema di Solovay

La **logica della dimostrabilità GL** (da Gödel-Löb) è il sistema modale che cattura esattamente il comportamento del predicato di dimostrabilità di PA. Il sistema è assiomatizzato da:

- Tutti i tautologie proposizionali
- K: $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
- GL (o principio di Löb): $\Box(\Box A \rightarrow A) \rightarrow \Box A$
- Regola di modus ponens

- Regola di necessitazione: se A è un teorema, allora $\Box A$ è un teorema

Il sistema GL è distinto dalla logica normale S4 (che aggiunge $\Box A \rightarrow \Box\Box A$ e $\Box A \rightarrow A$) e da S5. In GL, $\Box A \rightarrow A$ non è un teorema — ciò rispecchia il fatto che non tutto ciò che è dimostrabile in PA è vero (se PA è incoerente), e che in PA non si può dimostrare $\Box A \rightarrow A$ per ogni A (questo richiederebbe la coerenza di PA).

Il **teorema di completezza aritmetica di Solovay** (1976) è il risultato più profondo della logica della dimostrabilità: afferma che GL è completa rispetto all'interpretazione aritmetica, ossia per ogni formula A del linguaggio modale:

$$GL \vdash A \leftrightarrow \text{per ogni aritmetizzazione : } PA \vdash A^*$$

dove A^* è l'interpretazione di A in PA ottenuta sostituendo ogni variabile proposizionale p con una sentenza aritmetica e ogni $\Box B$ con $\text{Prov}(\ulcorner B^* \urcorner)$. In altri termini, le formule modali che PA può dimostrare per qualunque interpretazione delle variabili proposizionali sono esattamente i teoremi di GL.

La dimostrazione del teorema di Solovay è costruttiva: per ogni formula A non dimostrabile in GL, si costruisce esplicitamente un'interpretazione aritmetica A^* tale che $PA \not\vdash A^*$. La costruzione usa le proprietà di GL e le condizioni HBL in modo essenziale.

Il teorema di Solovay ha la conseguenza che la struttura completa della logica gödeliana — tutti i fatti dimostrabili sul predicato di dimostrabilità — è catturata dal sistema GL, e GL è decidibile. C'è dunque un senso preciso in cui sappiamo tutto ciò che può essere detto formalmente sulla dimostrabilità in PA, anche se PA stessa è incompleta: la metalogica è completa anche se la logica oggetto non lo è.

I.8 Il reverse mathematics: la struttura assiomatica dei teoremi ordinari

Il programma del **reverse mathematics**, sviluppato da Harvey Friedman a partire dal 1974 e sistematizzato da Stephen Simpson nel volume *Subsystems of Second Order Arithmetic* (1999), studia la forza assiomatica dei teoremi della matematica ordinaria determinando i sistemi assiomatici minimi necessari per dimostrarli.

Il framework di base è l'**aritmetica del secondo ordine** ($\mathcal{Z}_2$), che estende PA con variabili che spaziano su insiemi di numeri naturali e la comprensione degli insiemi. I cinque sottosistemi fondamentali ("the Big Five") sono:

RCA_0 (*Recursive Comprehension Axiom*): comprensione degli insiemi Δ_1 e induzione Σ_1 . Questo è il sistema base, corrispondente alla matematica computabile. In RCA_0 si dimostrano: i teoremi elementari dell'algebra lineare, il teorema di divisione, il teorema cinese del resto.

WKL_0 (*Weak König's Lemma*): RCA_0 più il lemma di König debole — ogni albero binario infinito ha un percorso infinito. WKL_0 corrisponde alla compattezza di Cantor e alle dimostrazioni per compattezza. Teoremi equivalenti a WKL_0 su RCA_0 : il teorema di Heine-Borel, il teorema di completezza di Gödel per la logica del primo ordine, il teorema di Brouwer per il punto fisso nello spazio discreto.

ACA_0 (*Arithmetic Comprehension Axiom*): RCA_0 più la comprensione degli insiemi aritmetici. Corrisponde all'analisi computabile con accesso all'oracolo per l'insieme dei teoremi di PA. Teoremi equivalenti su WKL_0 : il teorema di Bolzano-Weierstrass, il teorema della convergenza monotona, ogni sequenza di Cauchy dei numeri reali converge, il teorema di König per gli alberi finitamente ramificati.

ATR₀ (*Arithmetic Transfinite Recursion*): ACA₀ più la ricorsione transfinita su ordinali aritmetici. Corrisponde alla teoria dei ranghi nelle analisi e ai teoremi di separazione. Teoremi equivalenti su ACA₀: il teorema di Ulm per i gruppi abeliani, il teorema di Luzin sulla separazione delle analisi.

Π¹₁-CA₀ (*Π¹₁ Comprehension*): ATR₀ più la comprensione degli insiemi Π¹₁. Il sistema più forte tra i Big Five, corrispondente all'analisi predicativa impredicativa. Teoremi equivalenti su ATR₀: il teorema di Silver sul forcing, certi teoremi di Ramsey per gli insiemi infiniti.

Il risultato empirico notevole del programma è che quasi tutti i teoremi della matematica ordinaria — analisi, algebra, topologia, combinatoria — cadono in uno di questi cinque livelli, e la maggioranza cade nei tre livelli inferiori. Questo rivela la struttura gerarchica della forza dimostrativa: teoremi che sembrano ugualmente elementari richiedono assiomi di forza radicalmente diversa.

Un caso particolarmente elegante: il teorema di Ramsey per le triple — ogni 2-colorazione di $[\mathbb{N}]^3$ (le triple di naturali) ha un insieme omogeneo infinito — è equivalente ad ACA₀. Ma il teorema di Ramsey per le quadruple — ogni 2-colorazione di $[\mathbb{N}]^4$ ha un insieme omogeneo infinito — ha forza maggiore, richiedendo sistemi al di sopra di ACA₀. La forza assiomatica cresce con la dimensione delle uple, un fenomeno che non ha spiegazione intuitiva ovvia e che il reverse mathematics rivela in modo preciso.

Appendice Tecnica II

Dimensione VC, lemma di Sauer, bound di generalizzazione e geometria dell'informazione

II.1 La funzione di crescita e il lemma di Sauer

Per derivare i bound di generalizzazione nella loro forma quantitativa precisa, occorre introdurre la **funzione di crescita** di una classe di ipotesi, che misura il numero di dicotomie distinte che la classe può realizzare su un insieme finito.

Definizione: La funzione di crescita $\Pi_{\mathcal{H}}: \mathbb{N} \rightarrow \mathbb{N}$ di una classe di ipotesi binarie $\mathcal{H}$ è definita da:

$$\Pi_{\mathcal{H}}(m) = \max_{\{S \subset X, |S|=m\}} |\{(h(x_1), \dots, h(x_m)) : h \in \mathcal{H}\}|$$

ossia il massimo numero di classificazioni binarie distinte di un insieme di m punti che $\mathcal{H}$ può realizzare. $\Pi_{\mathcal{H}}(m) \leq 2^m$ per ogni m , con uguaglianza se e solo se $\mathcal{H}$ può spezzare qualche insieme di m punti.

Lemma di Sauer-Shelah (1972): Se $\text{VCdim}(\mathcal{H}) = d$, allora per ogni $m \geq d$:

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^d C(m, i) \leq (em/d)^d$$

dove $C(m, i) = m!/(i!(m-i)!)$ è il coefficiente binomiale e e è la base del logaritmo naturale. Per $m \geq 2d$, il bound si approssima a m^d (crescita polinomiale), a differenza del bound esponenziale 2^m .

Dimostrazione del lemma di Sauer: Per induzione su $m+d$. La base $m=0$ o $d=0$ è banale: $\sum_{i=0}^0 C(0, i) = 1 = \Pi_{\mathcal{H}}(0)$, e se $d=0$ allora $\mathcal{H}$ contiene un solo classificatore costante, quindi $\Pi_{\mathcal{H}}(m) = 1 = \sum_{i=0}^0 C(m, i)$.

Per il passo induttivo, si fissi un insieme $S = \{x_1, \dots, x_m\}$ e si considerino le dicotomie di $\mathcal{H}$ su S . Definiamo:

$$\mathcal{H}|_S = \{h(x_1), \dots, h(x_m)\} : h \in \mathcal{H}\}$$

e, per ogni sottoinsieme $S' = \{x_1, \dots, x_{m-1}\}$ di S privato dell'ultimo elemento:

$$\mathcal{H}|_{S'} = \{h(x_1), \dots, h(x_{m-1})\} : h \in \mathcal{H}\}$$

$$\mathcal{H}^+|_{S'} = \{h(x_1), \dots, h(x_{m-1})\} : \exists h \in \mathcal{H} \text{ con } h(x_m) = +1 \text{ e } \exists h' \in \mathcal{H} \text{ con } h'(x_m) = -1 \text{ e } h(x_i) = h'(x_i) \text{ per } i < m\}$$

ossia $\mathcal{H}^+|_{S'}$ è l'insieme delle prefissi di lunghezza $m-1$ di quelle dicotomie di S che compaiono sia con etichetta $+1$ che -1 su x_m . Allora:

$$|\mathcal{H}|_S| = |\mathcal{H}|_{S'}| + |\mathcal{H}^+|_{S'}|$$

La prima classe è una restrizione di $\mathcal{H}$ a S' , quindi $|\mathcal{H}|_{S'}| \leq \Pi_{\mathcal{H}}(m-1)$.

Per la seconda classe, si mostra che $VCdim(\mathcal{H}^+|_{S'}) \leq d-1$: ogni insieme spezzato da $\mathcal{H}^+|_{S'}$ può essere esteso con x_m a un insieme spezzato da $\mathcal{H}$, quindi la sua dimensione è al più $d-1$. Per ipotesi induttiva: $|\mathcal{H}^+|_{S'}| \leq \Pi_{\mathcal{H}^+}(m-1) \leq \sum_{i=0}^{d-1} C(m-1, i)$.

$$\text{Dunque: } |\mathcal{H}|_S| \leq \sum_{i=0}^{d-1} C(m-1, i) + \sum_{i=0}^{d-1} C(m-1, i)$$

Usando l'identità di Pascal $C(m, i) = C(m-1, i) + C(m-1, i-1)$:

$$\sum_{i=0}^{d-1} C(m-1, i) + \sum_{i=0}^{d-1} C(m-1, i) = \sum_{i=0}^d C(m, i) - C(m, d) + C(m-1, d-1) + C(m-1, d)$$

Un conto diretto mostra che il termine aggiuntivo è $C(m, d)$, e dunque

$$|\mathcal{H}|_S| \leq \sum_{i=0}^d C(m, i). \blacksquare$$

II.2 Il teorema di convergenza uniforme e il bound VC completo

Il bound di generalizzazione che è l'obiettivo di questa sezione è il seguente. Sia $\mathcal{H}$ una classe di ipotesi binarie con $VCdim(\mathcal{H}) = d < \infty$.

Allora per ogni distribuzione D su $X \times \{\pm 1\}$ e ogni $\delta > 0$, con probabilità almeno $1 - \delta$ rispetto al campionamento di un insieme di addestramento S di dimensione m , per ogni $h \in \mathcal{H}$:

$$\text{err}_D(h) \leq \text{err}_S(h) + \sqrt{(8d \log(2em/d)/m + 8 \log(4/\delta)/m)}$$

dove $\text{err}_S(h) = (1/m) \sum_i \mathbb{1}[h(x_i) \neq y_i]$ è l'errore empirico.

La dimostrazione procede in tre passi.

Passo 1: Doppia campionatura (symmetrization). Si sostituisce la distribuzione reale D con un *ghost sample* S' di dimensione m , mostrando che:

$$P_S[\sup_{h \in \mathcal{H}} (\text{err}_D(h) - \text{err}_S(h)) > \epsilon] \leq 2 P_{\{S, S'\}}[\sup_{h \in \mathcal{H}} (\text{err}_{\{S'\}}(h) - \text{err}_S(h)) > \epsilon/2]$$

purché $m \geq 2/\epsilon^2$. La simmetrizzazione riduce la probabilità sulla distribuzione a una probabilità su campioni finiti.

Passo 2: Permutazione. Per un insieme fisso di $2m$ punti $(S \cup S')$, le ipotesi $h \in \mathcal{H}$ realizzano al più $\Pi_{\mathcal{H}}(2m)$ classificazioni distinte. Dunque:

$$P_{\{\sigma\}}[\sup_{h \in \mathcal{H}} (\text{err}_{\{S'\}}(h) - \text{err}_S(h)) > \epsilon/2] \leq \Pi_{\mathcal{H}}(2m) \cdot P_{\{\sigma\}}[\text{err}_{\{S'\}}(h_0) - \text{err}_S(h_0) > \epsilon/2]$$

per una qualunque h_0 fissata, dove la probabilità è su permutazioni casuali σ che distribuiscono i $2m$ punti tra S e S' .

Passo 3: Bound di Chernoff. Per una h_0 fissata, la differenza $\text{err}_{\{S'\}}(h_0) - \text{err}_S(h_0)$ è la somma di m variabili i.i.d. limitate. Il bound di Hoeffding afferma:

$$P[\text{err}_{\{S'\}}(h_0) - \text{err}_S(h_0) > \epsilon/2] \leq \exp(-m\epsilon^2/8)$$

Combinando i tre passi con il lemma di Sauer:

$$P_S[\sup_{h \in \mathcal{H}} (\text{err}_D(h) - \text{err}_S(h)) > \epsilon] \leq 2 \Pi_{\mathcal{H}}(2m) \exp(-m\epsilon^2/8) \leq 2(2em/d)^d \cdot \exp(-m\epsilon^2/8)$$

Imponendo questa quantità uguale a δ e risolvendo per ϵ :

$$\epsilon = \sqrt{(8/m \cdot (d \log(2em/d) + \log(4/\delta)))}$$

che produce il bound dichiarato. ■

II.3 Il bound di Rademacher: forma precisa e connessione con VC

Il bound di Rademacher offre una stima della generalizzazione più fine del bound VC perché dipende dal campione osservato anziché solo dalla dimensione della classe. Riprendiamo la definizione dalla Parte B e ne dimostriamo il bound principale.

Teorema (Bartlett e Mendelson, 2002): Con probabilità almeno $1 - \delta$, per ogni $f \in \mathcal{F}$:

$$\text{err}_D(f) \leq \text{err}_S(f) + 2 R_S(\mathcal{F}) + \sqrt{(2 \log(1/\delta))/m}$$

Dimostrazione: Si definisce $\Phi(S) = \sup_{f \in \mathcal{F}} (\text{err}_D(f) - \text{err}_S(f))$. Questo è una funzione stabile sotto sostituzione di un singolo elemento del campione (cambio di un punto (x_i, y_i) modifica Φ di al più $1/m$), quindi per il **lemma di McDiarmid**:

$$P[\Phi(S) > E[\Phi(S)] + t] \leq \exp(-2mt^2)$$

Prendendo $t = \sqrt{(\log(1/\delta)/(2m))}$: con probabilità $1 - \delta$, $\Phi(S) \leq E[\Phi(S)] + \sqrt{(\log(1/\delta)/(2m))}$.

Rimane da bound $E[\Phi(S)]$. Per il trucco della simmetrizzazione con variabili di Rademacher $\sigma_1, \dots, \sigma_n$ i.i.d. uniformi su $\{\pm 1\}$:

$$\begin{aligned} E_S[\Phi(S)] &= E_S[\sup_f E_{\{S'\}}[\text{err}_{\{S'\}}(f) - \text{err}_S(f)] \leq \\ E_{\{S,S'\}}[\sup_f (\text{err}_{\{S'\}}(f) - \text{err}_S(f))] &= E_{\{S,S',\sigma\}}[\sup_f (1/m) \sum_i \sigma_i \\ (\mathbb{1}[f(x_i) \neq y_i] - \mathbb{1}[f(x_i) \neq y_i])] &\leq 2 E_{\{S,\sigma\}}[\sup_f (1/m) \sum_i \sigma_i \mathbb{1}[f(x_i) \neq y_i]] = \\ 2 R_m(\mathcal{F} \circ \ell) \end{aligned}$$

dove l'ultima uguaglianza è la definizione di complessità di Rademacher della classe $\mathcal{F}$ composta con la funzione di perdita ℓ . ■

La **connessione con la dimensione VC**: si può mostrare che $R_m(\mathcal{H}) \leq \sqrt{(2d \log(em/d)/m)}$, quindi il bound di Rademacher implica il bound VC, ma è più fine quando la struttura del campione S permette di bound $R_S(\mathcal{H})$ in modo più stretto del worst-case.

II.4 La geometria dell'informazione: la varietà statistica

Il calcolo differenziale su famiglie di distribuzioni di probabilità è il cuore della **geometria dell'informazione**, sviluppata da Shun-ichi Amari a partire dagli anni Sessanta e sistematizzata nel volume *Differential-Geometrical Methods in Statistics* (1985). Questa teoria è il contesto naturale per comprendere il gradiente naturale e la sua connessione con la selezione naturale discussa nella Parte Quarta.

Una **famiglia statistica** è un insieme di distribuzioni di probabilità parametrizzate:

$$\mathcal{M} = \{p(x; \theta) : \theta \in \Theta \subseteq \mathbb{R}^n\}$$

che soddisfa condizioni di regolarità (differenziabilità in θ , scambio di integrazione e derivazione). La famiglia è una **varietà statistica** — una varietà differenziabile n -dimensionale — dove θ sono le coordinate locali e le distribuzioni sono i punti.

La struttura metrica sulla varietà è fornita dalla **metrica di Fisher**:

$$g_{ij}(\theta) = E_{x \sim p(\cdot; \theta)} [\partial_i \log p(x; \theta) \cdot \partial_j \log p(x; \theta)]$$

dove $\partial_i = \partial / \partial \theta_i$. Questa è la **matrice di informazione di Fisher** (FIM). Il tensore $g_{ij}(\theta)$ è semidefinito positivo per ogni θ , e definisce un prodotto interno sullo spazio tangente $T_{\theta} \mathcal{M}$.

La metrica di Fisher ha una caratterizzazione invariante importante: è l'unica metrica (a meno di scalatura) che è invariante rispetto alle trasformazioni sufficienti dei dati. Se T è una statistica sufficiente — se la distribuzione condizionale di x dato $T(x)$ non dipende da θ — allora la variante di Fisher dopo trasformazione attraverso T è uguale a quella prima. Questa invarianza la rende la metrica "canonica" per la famiglia.

II.5 La divergenza KL come distanza sulla varietà statistica

La **divergenza di Kullback-Leibler** tra due distribuzioni nella stessa famiglia è:

$$D_{\text{KL}}(p(\cdot; \theta) \parallel p(\cdot; \theta')) = E_{x \sim p(\cdot; \theta)} [\log p(x; \theta) - \log p(x; \theta')] = \int p(x; \theta) \log [p(x; \theta) / p(x; \theta')] dx$$

La KL non è una distanza nel senso metrico (non è simmetrica e non soddisfa la disuguaglianza triangolare), ma è localmente quadratica in $(\theta - \theta')$ con il coefficiente determinato dalla FIM. Precisamente, per $\theta' = \theta + d\theta$:

$$\begin{aligned} D_{\text{KL}}(p(\cdot; \theta) \parallel p(\cdot; \theta + d\theta)) &= (1/2) \sum_{i,j} g_{ij}(\theta) d\theta_i d\theta_j + O(|d\theta|^3) \\ &= (1/2) d\theta^T G(\theta) d\theta + O(|d\theta|^3) \end{aligned}$$

dove $G(\theta)$ è la FIM come matrice. Questo mostra che la FIM è la forma quadratica che approssima la KL a secondo ordine: il "costo" di muoversi dal punto θ a $\theta + d\theta$ sulla varietà statistica è approssimato dalla metrica di Fisher.

II.6 Il gradiente naturale e la sua derivazione

Nel contesto dell'apprendimento statistico, si ha una funzione obiettivo $L(\theta)$ da minimizzare (la perdita attesa) su una varietà statistica $\mathcal{M}$. Il **gradiente standard** $\nabla_{\theta} L$ è un vettore nello spazio dei parametri $\mathbb{R}^n$ che punta nella direzione di massima salita della perdita nelle coordinate euclidee. Ma le coordinate θ non hanno struttura euclidea canonica: la stessa variazione $d\theta$ corrisponde a spostamenti diversi sulla varietà statistica a seconda della curvatura locale.

Il **gradiente naturale** di Amari è il gradiente rispetto alla metrica di Fisher:

$$\nabla^{\sim}_{\theta} L = G(\theta)^{-1} \nabla_{\theta} L$$

dove $G(\theta) = [g_{ij}(\theta)]$ è la FIM come matrice $n \times n$. Il gradiente naturale è il vettore che punta nella direzione di massima salita di L rispetto alla metrica di Fisher — ossia rispetto alla geometria della varietà statistica, non alla geometria euclidea delle coordinate.

La derivazione formale: sulla varietà con metrica G , il gradiente è definito come il vettore v che soddisfa $dL = \langle v, \cdot \rangle_G$, ossia $dL(w) = v^T G w$ per ogni w . Quindi $v^T G w = \nabla_{\theta} L \cdot w$ per ogni w , che dà $G v = \nabla_{\theta} L$, ossia $v = G^{-1} \nabla_{\theta} L$.

La regola di aggiornamento del gradiente naturale è:

$$\theta \leftarrow \theta - \eta G(\theta)^{-1} \nabla_{\theta} L$$

e ha proprietà di invarianza che la SGD euclidea non ha: l'aggiornamento produce lo stesso spostamento sul manifold statistico indipendentemente dalla parametrizzazione scelta.

II.7 La connessione con la selezione naturale

Il parallelismo formale tra selezione naturale e gradiente naturale, evocato nella Parte Quarta, può essere reso preciso nel seguente modo. Nella genetica delle popolazioni, le variabili θ sono le frequenze alleliche in un locus multi-allelico: $\theta = (p_1, \dots, p_k)$ con $\sum_i p_i = 1$ — un punto nel simpleso statistico.

La distribuzione di probabilità associata è la distribuzione multinomiale: un individuo campionato ha allele i con probabilità p_i . La FIM della distribuzione multinomiale nel simpleso è la **metrica di Shahshahani**:

$$g_{ij}(\mathbf{p}) = \delta_{ij} / p_i \quad (\text{in coordinate locali sul simpleso})$$

— la metrica che rende il simpleso una sottovarietà di raggio 2 della sfera in $\mathbb{R}^k$ attraverso la trasformazione $\theta_i = 2\sqrt{p_i}$.

La funzione obiettivo dell'evoluzione è la fitness media $\bar{w}(\mathbf{p}) = \sum_i p_i w_i$, dove w_i è la fitness dell'allele i . Il **teorema fondamentale della selezione naturale di Fisher** afferma che la variazione della fitness media per generazione è uguale alla varianza adattiva:

$$d\bar{w}/dt = \sum_{i,j} \{i,j\} g_{ij}(\mathbf{p}) \cdot \partial \bar{w} / \partial p_i \cdot \partial \bar{w} / \partial p_j = \|\nabla_{\mathbf{G}} \bar{w}\|_{\mathbf{G}}^2$$

Ossia: la selezione naturale fa crescere la fitness nella direzione del gradiente naturale della fitness rispetto alla metrica di Shahshahani. L'equazione di selezione di Fisher:

$$dp_i/dt = p_i (w_i - \bar{w})$$

è precisamente la regola di aggiornamento del gradiente naturale della fitness sulla varietà multinomiale:

$$dp_i/dt = \sum_j G^{-1}_{ij} \cdot \partial \bar{w} / \partial p_j$$

Una verifica: $G^{-1}_{ij} = p_i \delta_{ij} - p_i p_j$ (l'inverso della metrica di Shahshahani sul simpleso). Dunque:

$$\sum_i G^{-1}_{ii} \cdot \partial \bar{w} / \partial p_i = \sum_i (p_i \delta_{ii} - p_i p_i) w_i = p_i w_i - p_i \sum_i p_i w_i = p_i (w_i - \bar{w}) \quad \checkmark$$

La selezione naturale è pertanto, in senso matematicamente preciso, gradient ascent della fitness rispetto alla metrica di informazione di Fisher sulla varietà delle distribuzioni di frequenza genica. Questo è il contenuto esatto dell'affermazione nella Parte Quarta che selezione naturale e gradient descent sono istanze dello stesso algoritmo su varietà riemanniane con metrica determinata dalla FIM.

II.8 La ELBO e la derivazione completa del bound variazionale

La **Evidence Lower BOund** (ELBO) è il funzionale centrale dell'inferenza variazionale e della codifica predittiva di Friston. La sua derivazione completa chiarisce la connessione tra i due.

Sia x il dato osservato e z la variabile latente. Il modello generativo è $P(x, z) = P(x|z) P(z)$. L'obiettivo bayesiano è calcolare la posteriore $P(z|x) = P(x, z)/P(x)$, dove $P(x) = \int P(x|z) P(z) dz$ è l'evidenza del modello — l'integrale marginale che è in generale intrattabile.

L'inferenza variazionale approssima $P(z|x)$ con una distribuzione $Q(z; \lambda)$ da una famiglia trattabile, minimizzando la divergenza KL:

$$D_{\{KL\}}(Q(z; \lambda) \parallel P(z|x)) = E_Q[\log Q(z; \lambda) - \log P(z|x)] = E_Q[\log Q(z; \lambda) - \log P(x, z) + \log P(x)] = -E_Q[\log P(x, z) - \log Q(z; \lambda)] + \log P(x) = -\mathcal{L}(\lambda) + \log P(x)$$

dove $\mathcal{L}(\lambda) = E_Q[\log P(x, z) - \log Q(z; \lambda)]$ è la ELBO. Poiché $D_{\{KL\}} \geq 0$, otteniamo:

$$\log P(x) \geq \mathcal{L}(\lambda) = E_Q[\log P(x|z)] - D_{\{KL\}}(Q(z; \lambda) \parallel P(z))$$

La ELBO è scomponibile in due termini:

(i) $E_Q[\log P(x|z)]$: il termine di ricostruzione — la verosimiglianza attesa dei dati sotto il modello generativo, pesata dalla distribuzione variazionale. Misura quanto il modello è capace di ricostruire l'osservazione.

(ii) $-D_{\text{KL}}(Q(z;\lambda) \parallel P(z))$: il termine di regolarizzazione — il negativo della KL tra la distribuzione variazionale e il prior. Penalizza distribuzioni variazionali lontane dal prior, inducendo semplicità nelle rappresentazioni latenti.

Massimizzare la ELBO equivale simultaneamente a minimizzare la KL tra Q e la vera posteriore (minimizzando l'approssimazione) e a massimizzare il log-evidence del modello (adattando il modello ai dati). Per la codifica predittiva di Friston, il sistema cognitivo minimizza $F = -\mathcal{L}$ — l'energia libera variazionale — simultaneamente rispetto ai parametri del modello (apprendimento) e rispetto alle variabili di stato (percezione).

II.9 La stima del gradiente della ELBO: il trucco della riparametrizzazione

Per ottimizzare la ELBO rispetto ai parametri λ della distribuzione variazionale $Q(z; \lambda)$, occorre calcolare $\nabla_{\lambda} \mathcal{L}(\lambda)$. Questo richiede calcolare il gradiente di un'aspettazione rispetto a una distribuzione che dipende dai parametri — operazione che il gradiente standard non permette direttamente.

Il **trucco della riparametrizzazione** di Kingma e Welling (2014) risolve questo problema per distribuzioni per cui $z \sim Q(z; \lambda)$ può essere scritto come $z = g(\epsilon, \lambda)$ dove $\epsilon \sim p(\epsilon)$ è una variabile di rumore con distribuzione fissa (indipendente da λ) e g è una funzione differenziabile.

Per la gaussiana $Q(z; \mu, \sigma^2) = \mathcal{N}(\mu, \sigma^2)$, la riparametrizzazione è $z = \mu + \sigma \epsilon$ con $\epsilon \sim \mathcal{N}(0,1)$.

Sotto questa riparametrizzazione:

$$\nabla_{\lambda} \mathcal{L}(\lambda) = \nabla_{\lambda} E_{\{\epsilon \sim p(\epsilon)\}} [\log P(x, g(\epsilon, \lambda)) - \log Q(g(\epsilon, \lambda); \lambda)] = E_{\{\epsilon \sim p(\epsilon)\}} [\nabla_{\lambda} (\log P(x, g(\epsilon, \lambda)) - \log Q(g(\epsilon, \lambda); \lambda))]$$

Il gradiente passa sotto il segno di aspettazione perché $p(\epsilon)$ è indipendente da λ : il risultato è una media di gradienti che può essere stimata con campioni Monte Carlo. Questo produce lo **stimatore del gradiente della ELBO** per il Variational Autoencoder (VAE):

$$\nabla_{\lambda} \mathcal{L} \approx (1/L) \sum_{l=1}^L \nabla_{\lambda} [\log P(x | g(\epsilon^l, \lambda)) - \log Q(g(\epsilon^l, \lambda); \lambda) + \log P(g(\epsilon^l, \lambda))]$$

con $\epsilon^l \sim p(\epsilon)$ campionati i.i.d. La varianza di questo stimatore è molto inferiore a quella dello stimatore REINFORCE (che non usa la riparametrizzazione), rendendo l'addestramento stabile.

Dimensione VC, lemma di Sauer, bound di generalizzazione e geometria dell'informazione

II.1 La funzione di crescita e il lemma di Sauer

Per derivare i bound di generalizzazione nella loro forma quantitativa precisa, occorre introdurre la **funzione di crescita** di una classe di ipotesi, che misura il numero di dicotomie distinte che la classe può realizzare su un insieme finito.

Definizione: La funzione di crescita $\Pi_{\mathcal{H}}: \mathbb{N} \rightarrow \mathbb{N}$ di una classe di ipotesi binarie $\mathcal{H}$ è definita da:

$$\Pi_{\mathcal{H}}(m) = \max_{\{S \subset X, |S|=m\}} |\{(h(x_1), \dots, h(x_m)) : h \in \mathcal{H}\}|$$

ossia il massimo numero di classificazioni binarie distinte di un insieme di m punti che $\mathcal{H}$ può realizzare. $\Pi_{\mathcal{H}}(m) \leq 2^m$ per ogni m , con uguaglianza se e solo se $\mathcal{H}$ può spezzare qualche insieme di m punti.

Lemma di Sauer-Shelah (1972): Se $VCdim(\mathcal{H}) = d$, allora per ogni $m \geq d$:

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^d C(m, i) \leq (em/d)^d$$

dove $C(m, i) = m! / (i!(m-i)!)$ è il coefficiente binomiale e e è la base del logaritmo naturale. Per $m \geq 2d$, il bound si approssima a m^d (crescita polinomiale), a differenza del bound esponenziale 2^m .

Dimostrazione del lemma di Sauer: Per induzione su $m+d$. La base $m=0$ o $d=0$ è banale: $\sum_{i=0}^0 C(0, i) = 1 = \Pi_{\mathcal{H}}(0)$, e se $d=0$ allora $\mathcal{H}$ contiene un solo classificatore costante, quindi $\Pi_{\mathcal{H}}(m) = 1 = \sum_{i=0}^0 C(m, i)$.

Per il passo induttivo, si fissi un insieme $S = \{x_1, \dots, x_m\}$ e si considerino le dicotomie di $\mathcal{H}$ su S . Definiamo:

$$\mathcal{H}|_S = \{(h(x_1), \dots, h(x_m)) : h \in \mathcal{H}\}$$

e, per ogni sottoinsieme $S' = \{x_1, \dots, x_{m-1}\}$ di S privato dell'ultimo elemento:

$$\mathcal{H}|_{S'} = \{(h(x_1), \dots, h(x_{m-1})) : h \in \mathcal{H}\}$$

$$\mathcal{H}^+|_{S'} = \{(h(x_1), \dots, h(x_{m-1})) : \exists h \in \mathcal{H} \text{ con } h(x_m) = +1 \text{ e } \exists h' \in \mathcal{H} \text{ con } h'(x_m) = -1 \text{ e } h(x_i) = h'(x_i) \text{ per } i < m\}$$

ossia $\mathcal{H}^+|_{S'}$ è l'insieme delle prefissi di lunghezza $m-1$ di quelle dicotomie di S che compaiono sia con etichetta $+1$ che -1 su x_m . Allora:

$$|\mathcal{H}|_S| = |\mathcal{H}|_{S'}| + |\mathcal{H}^+|_{S'}|$$

La prima classe è una restrizione di $\mathcal{H}$ a S' , quindi $|\mathcal{H}|_{S'}| \leq \Pi_{\mathcal{H}}(m-1)$. Per la seconda classe, si mostra che $VCdim(\mathcal{H}^+|_{S'}) \leq d-1$: ogni insieme spezzato da $\mathcal{H}^+|_{S'}$ può essere esteso con x_m a un insieme spezzato da $\mathcal{H}$, quindi la sua

dimensione è al più $d-1$. Per ipotesi induttiva: $|\mathcal{H}^+ \mid \{S'\}| \leq \Pi_{-}\{\mathcal{H}^+\}(m-1) \leq \sum_{i=0}^{d-1} C(m-1, i)$.

Dunque: $|\mathcal{H} \mid_{-S}| \leq \sum_{i=0}^{d-1} C(m-1, i) + \sum_{i=0}^{d-1} C(m-1, i)$

Usando l'identità di Pascal $C(m, i) = C(m-1, i) + C(m-1, i-1)$:

$$\sum_{i=0}^{d-1} C(m-1, i) + \sum_{i=0}^{d-1} C(m-1, i) = \sum_{i=0}^d C(m, i) - C(m, d) + C(m-1, d-1) + C(m-1, d)$$

Un conto diretto mostra che il termine aggiuntivo è $C(m, d)$, e dunque $|\mathcal{H} \mid_{-S}| \leq \sum_{i=0}^d C(m, i)$. ■

II.2 Il teorema di convergenza uniforme e il bound VC completo

Il bound di generalizzazione che è l'obiettivo di questa sezione è il seguente. Sia $\mathcal{H}$ una classe di ipotesi binarie con $VCdim(\mathcal{H}) = d < \infty$. Allora per ogni distribuzione D su $X \times \{\pm 1\}$ e ogni $\delta > 0$, con probabilità almeno $1 - \delta$ rispetto al campionamento di un insieme di addestramento S di dimensione m , per ogni $h \in \mathcal{H}$:

$$err_D(h) \leq err_S(h) + \sqrt{(8d \log(2em/d)/m + 8 \log(4/\delta)/m)}$$

dove $err_S(h) = (1/m) \sum_i \mathbb{1}[h(x_i) \neq y_i]$ è l'errore empirico.

La dimostrazione procede in tre passi.

Passo 1: Doppia campionatura (symmetrization). Si sostituisce la distribuzione reale D con un ghost sample S' di dimensione m , mostrando che:

$$P_S[\sup_{h \in \mathcal{H}} (err_D(h) - err_S(h)) > \varepsilon] \leq 2 P_{\{S, S'\}}[\sup_{h \in \mathcal{H}} (err_{\{S'\}}(h) - err_S(h)) > \varepsilon/2]$$

purché $m \geq 2/\varepsilon^2$. La simmetrizzazione riduce la probabilità sulla distribuzione a una probabilità su campioni finiti.

Passo 2: Permutazione. Per un insieme fisso di $2m$ punti $(S \cup S')$, le ipotesi $h \in \mathcal{H}$ realizzano al più $\Pi_{-}\mathcal{H}(2m)$ classificazioni distinte. Dunque:

$$P_{\{\sigma\}}[\sup_{\{h \in \mathcal{H}\}} (err_{S'}(h) - err_S(h)) > \varepsilon/2] \leq \Pi_{\mathcal{H}}(2m) \cdot P_{\{\sigma\}}[err_{S'}(h_0) - err_S(h_0) > \varepsilon/2]$$

per una qualunque h_0 fissata, dove la probabilità è su permutazioni casuali σ che distribuiscono i $2m$ punti tra S e S' .

Passo 3: Bound di Chernoff. Per una h_0 fissata, la differenza $err_{S'}(h_0) - err_S(h_0)$ è la somma di m variabili i.i.d. limitate. Il bound di Hoeffding afferma:

$$P[err_{S'}(h_0) - err_S(h_0) > \varepsilon/2] \leq \exp(-m\varepsilon^2/8)$$

Combinando i tre passi con il lemma di Sauer:

$$P_S[\sup_{\{h \in \mathcal{H}\}} (err_D(h) - err_S(h)) > \varepsilon] \leq 2 \Pi_{\mathcal{H}}(2m) \exp(-m\varepsilon^2/8) \leq 2(2em/d)^d \cdot \exp(-m\varepsilon^2/8)$$

Imponendo questa quantità uguale a δ e risolvendo per ε :

$$\varepsilon = \sqrt{(8/m \cdot (d \log(2em/d) + \log(4/\delta)))}$$

che produce il bound dichiarato. ■

II.3 Il bound di Rademacher: forma precisa e connessione con VC

Il bound di Rademacher offre una stima della generalizzazione più fine del bound VC perché dipende dal campione osservato anziché solo dalla dimensione della classe. Riprendiamo la definizione dalla Parte B e ne dimostriamo il bound principale.

Teorema (Bartlett e Mendelson, 2002): Con probabilità almeno $1 - \delta$, per ogni $f \in \mathcal{F}$:

$$err_D(f) \leq err_S(f) + 2 R_S(\mathcal{F}) + \sqrt{(2 \log(1/\delta))/m}$$

Dimostrazione: Si definisce $\Phi(S) = \sup_{\{f \in \mathcal{F}\}} (err_D(f) - err_S(f))$. Questo è una funzione stabile sotto sostituzione di un singolo elemento del campione (cambio di un punto (x_i, y_i) modifica Φ di al più $1/m$), quindi per il **lemma di McDiarmid**:

$$P[\Phi(S) > E[\Phi(S)] + t] \leq \exp(-2mt^2)$$

Prendendo $t = \sqrt{(\log(1/\delta)/(2m))}$: con probabilità $1 - \delta$, $\Phi(S) \leq E[\Phi(S)] + \sqrt{(\log(1/\delta)/(2m))}$.

Rimane da bound $E[\Phi(S)]$. Per il trucco della simmetrizzazione con variabili di Rademacher $\sigma_1, \dots, \sigma$ i.i.d. uniformi su $\{\pm 1\}$:

$$\begin{aligned} E_S[\Phi(S)] &= E_S[\sup_f E_{\{S'\}}[err_{\{S'\}}(f) - err_S(f)]] \leq E_{\{S, S'\}}[\sup_f (err_{\{S'\}}(f) - err_S(f))] \\ &= E_{\{S, S', \sigma\}}[\sup_f (1/m) \sum_i \sigma_i (\mathbb{1}[f(x_i') \neq y_i'] - \mathbb{1}[f(x_i) \neq y_i])] \leq 2 E_{\{S, \sigma\}}[\sup_f (1/m) \sum_i \sigma_i \mathbb{1}[f(x_i) \neq y_i]] = 2 R_m(\mathcal{F} \circ \ell) \end{aligned}$$

dove l'ultima uguaglianza è la definizione di complessità di Rademacher della classe $\mathcal{F}$ composta con la funzione di perdita ℓ . ■

La **connessione con la dimensione VC**: si può mostrare che $R_m(\mathcal{H}) \leq \sqrt{(2d \log(em/d)/m)}$, quindi il bound di Rademacher implica il bound VC, ma è più fine quando la struttura del campione S permette di bound $R_S(\mathcal{H})$ in modo più stretto del worst-case.

II.4 La geometria dell'informazione: la varietà statistica

Il calcolo differenziale su famiglie di distribuzioni di probabilità è il cuore della **geometria dell'informazione**, sviluppata da Shun-ichi Amari a partire dagli anni Sessanta e sistematizzata nel volume *Differential-Geometrical Methods in Statistics* (1985). Questa teoria è il contesto naturale per comprendere il gradiente naturale e la sua connessione con la selezione naturale discussa nella Parte Quarta.

Una **famiglia statistica** è un insieme di distribuzioni di probabilità parametrizzate:

$$\mathcal{M} = \{p(x; \theta) : \theta \in \Theta \subseteq \mathbb{R}^n\}$$

che soddisfa condizioni di regolarità (differenziabilità in θ , scambio di integrazione e derivazione). La famiglia è una **varietà statistica** — una varietà differenziabile n -dimensionale — dove θ sono le coordinate locali e le distribuzioni sono i punti.

La struttura metrica sulla varietà è fornita dalla **metrica di Fisher**:

$$g_{ij}(\theta) = E_{x \sim p(\cdot; \theta)} [\partial_i \log p(x; \theta) \cdot \partial_j \log p(x; \theta)]$$

dove $\partial_i = \partial / \partial \theta_i$. Questa è la **matrice di informazione di Fisher** (FIM). Il tensore $g_{ij}(\theta)$ è semidefinito positivo per ogni θ , e definisce un prodotto interno sullo spazio tangente $T_\theta \mathcal{M}$.

La metrica di Fisher ha una caratterizzazione invariante importante: è l'unica metrica (a meno di scalatura) che è invariante rispetto alle trasformazioni sufficienti dei dati. Se T è una statistica sufficiente — se la distribuzione condizionale di x dato $T(x)$ non dipende da θ — allora la variante di Fisher dopo trasformazione attraverso T è uguale a quella prima. Questa invarianza la rende la metrica "canonica" per la famiglia.

II.5 La divergenza KL come distanza sulla varietà statistica

La **divergenza di Kullback-Leibler** tra due distribuzioni nella stessa famiglia è:

$$D_{\text{KL}}(p(\cdot; \theta) \parallel p(\cdot; \theta')) = E_{x \sim p(\cdot; \theta)} [\log p(x; \theta) - \log p(x; \theta')] = \int p(x; \theta) \log [p(x; \theta) / p(x; \theta')] dx$$

La KL non è una distanza nel senso metrico (non è simmetrica e non soddisfa la disuguaglianza triangolare), ma è localmente quadratica in $(\theta - \theta')$ con il coefficiente determinato dalla FIM. Precisamente, per $\theta' = \theta + d\theta$:

$$\begin{aligned} D_{\text{KL}}(p(\cdot; \theta) \parallel p(\cdot; \theta + d\theta)) &= (1/2) \sum_{i,j} g_{ij}(\theta) d\theta_i d\theta_j + O(|d\theta|^3) \\ &= (1/2) d\theta^T G(\theta) d\theta + O(|d\theta|^3) \end{aligned}$$

dove $G(\theta)$ è la FIM come matrice. Questo mostra che la FIM è la forma quadratica che approssima la KL a secondo ordine: il "costo" di muoversi dal punto θ a $\theta + d\theta$ sulla varietà statistica è approssimato dalla metrica di Fisher.

II.6 Il gradiente naturale e la sua derivazione

Nel contesto dell'apprendimento statistico, si ha una funzione obiettivo $L(\theta)$ da minimizzare (la perdita attesa) su una varietà statistica $\mathcal{M}$. Il **gradiente standard** $\nabla_{\theta} L$ è un vettore nello spazio dei parametri $\mathbb{R}^n$ che punta nella direzione di massima salita della perdita nelle coordinate euclidee. Ma le coordinate θ non hanno struttura euclidea canonica: la stessa variazione $d\theta$ corrisponde a spostamenti diversi sulla varietà statistica a seconda della curvatura locale.

Il **gradiente naturale** di Amari è il gradiente rispetto alla metrica di Fisher:

$$\nabla_{\sim \theta} L = G(\theta)^{-1} \nabla_{\theta} L$$

dove $G(\theta) = [g_{ij}(\theta)]$ è la FIM come matrice $n \times n$. Il gradiente naturale è il vettore che punta nella direzione di massima salita di L rispetto alla metrica di Fisher — ossia rispetto alla geometria della varietà statistica, non alla geometria euclidea delle coordinate.

La derivazione formale: sulla varietà con metrica G , il gradiente è definito come il vettore v che soddisfa $dL = \langle v, \cdot \rangle_G$, ossia $dL(w) = v^T G w$ per ogni w . Quindi $v^T G w = \nabla_{\theta} L \cdot w$ per ogni w , che dà $G v = \nabla_{\theta} L$, ossia $v = G^{-1} \nabla_{\theta} L$.

La regola di aggiornamento del gradiente naturale è:

$$\theta \leftarrow \theta - \eta G(\theta)^{-1} \nabla_{\theta} L$$

e ha proprietà di invarianza che la SGD euclidea non ha: l'aggiornamento produce lo stesso spostamento sul manifold statistico indipendentemente dalla parametrizzazione scelta.

II.7 La connessione con la selezione naturale

Il parallelismo formale tra selezione naturale e gradiente naturale, evocato nella Parte Quarta, può essere reso preciso nel seguente modo. Nella genetica delle popolazioni, le variabili θ sono le frequenze alleliche in un locus multi-allelico: $\theta = (p_1, \dots, p_k)$ con $\sum_i p_i = 1$ — un punto nel simpleso statistico.

La distribuzione di probabilità associata è la distribuzione multinomiale: un individuo campionato ha allele i con probabilità p_i . La FIM della distribuzione multinomiale nel simpleso è la **metrica di Shahshahani**:

$$g_{ij}(\mathbf{p}) = \delta_{ij} / p_i \quad (\text{in coordinate locali sul simpleso})$$

— la metrica che rende il simpleso una sottovarietà di raggio 2 della sfera in $\mathbb{R}^k$ attraverso la trasformazione $\theta_i = 2\sqrt{p_i}$.

La funzione obiettivo dell'evoluzione è la fitness media $\bar{w}(\mathbf{p}) = \sum_i p_i w_i$, dove w_i è la fitness dell'allele i . Il **teorema fondamentale della selezione naturale di Fisher** afferma che la variazione della fitness media per generazione è uguale alla varianza adattiva:

$$d\bar{w}/dt = \sum_{i,j} \{i,j\} g_{ij}(\mathbf{p}) \cdot \partial \bar{w} / \partial p_i \cdot \partial \bar{w} / \partial p_j = \|\nabla_{\mathbf{p}} \bar{w}\|_{\mathbf{g}}^2$$

Ossia: la selezione naturale fa crescere la fitness nella direzione del gradiente naturale della fitness rispetto alla metrica di Shahshahani. L'equazione di selezione di Fisher:

$$dp_i/dt = p_i (w_i - \bar{w})$$

è precisamente la regola di aggiornamento del gradiente naturale della fitness sulla varietà multinomiale:

$$dp_i/dt = \sum_j G^{-1}_{ij} \cdot \partial \bar{w} / \partial p_j$$

Una verifica: $G^{-1}_{ij} = p_i \delta_{ij} - p_i p_j$ (l'inverso della metrica di Shahshahani sul simpleso). Dunque:

$$\sum_j G^{-1}_{ij} \cdot \partial \bar{w} / \partial p_j = \sum_j (p_i \delta_{ij} - p_i p_j) w_j = p_i w_i - p_i \sum_j p_j w_j = p_i (w_i - \bar{w}) \quad \checkmark$$

La selezione naturale è pertanto, in senso matematicamente preciso, gradient ascent della fitness rispetto alla metrica di informazione di Fisher sulla varietà delle distribuzioni di frequenza genica. Questo è il contenuto esatto dell'affermazione nella Parte Quarta che selezione naturale e gradient descent sono istanze dello stesso algoritmo su varietà riemanniane con metrica determinata dalla FIM.

II.8 La ELBO e la derivazione completa del bound variazionale

La **Evidence Lower BOund** (ELBO) è il funzionale centrale dell'inferenza variazionale e della codifica predittiva di Friston. La sua derivazione completa chiarisce la connessione tra i due.

Sia x il dato osservato e z la variabile latente. Il modello generativo è $P(x, z) = P(x|z)P(z)$. L'obiettivo bayesiano è calcolare la posteriore $P(z|x) = P(x,z)/P(x)$, dove $P(x) = \int P(x|z)P(z) dz$ è l'evidenza del modello — l'integrale marginale che è in generale intrattabile.

L'inferenza variazionale approssima $P(z|x)$ con una distribuzione $Q(z; \lambda)$ da una famiglia trattabile, minimizzando la divergenza KL:

$$D_{\{KL\}}(Q(z; \lambda) \parallel P(z|x)) = E_Q[\log Q(z; \lambda) - \log P(z|x)] = E_Q[\log Q(z; \lambda) - \log P(x, z) + \log P(x)] = -E_Q[\log P(x, z) - \log Q(z; \lambda)] + \log P(x) = -\mathcal{L}(\lambda) + \log P(x)$$

dove $\mathcal{L}(\lambda) = E_Q[\log P(x, z) - \log Q(z; \lambda)]$ è la ELBO. Poiché $D_{\{KL\}} \geq 0$, otteniamo:

$$\log P(x) \geq \mathcal{L}(\lambda) = E_Q[\log P(x|z)] - D_{\{KL\}}(Q(z; \lambda) \parallel P(z))$$

La ELBO è scomponibile in due termini:

(i) $E_Q[\log P(x|z)]$: il termine di ricostruzione — la verosimiglianza attesa dei dati sotto il modello generativo, pesata dalla distribuzione variazionale. Misura quanto il modello è capace di ricostruire l'osservazione.

(ii) $-D_{\text{KL}}(Q(z;\lambda) \parallel P(z))$: il termine di regolarizzazione — il negativo della KL tra la distribuzione variazionale e il prior. Penalizza distribuzioni variazionali lontane dal prior, inducendo semplicità nelle rappresentazioni latenti.

Massimizzare la ELBO equivale simultaneamente a minimizzare la KL tra Q e la vera posteriore (minimizzando l'approssimazione) e a massimizzare il log-evidence del modello (adattando il modello ai dati). Per la codifica predittiva di Friston, il sistema cognitivo minimizza $F = -\mathcal{L}$ — l'energia libera variazionale — simultaneamente rispetto ai parametri del modello (apprendimento) e rispetto alle variabili di stato (percezione).

II.9 La stima del gradiente della ELBO: il trucco della riparametrizzazione

Per ottimizzare la ELBO rispetto ai parametri λ della distribuzione variazionale $Q(z; \lambda)$, occorre calcolare $\nabla_{\lambda} \mathcal{L}(\lambda)$. Questo richiede calcolare il gradiente di un'aspettazione rispetto a una distribuzione che dipende dai parametri — operazione che il gradiente standard non permette direttamente.

Il **trucco della riparametrizzazione** di Kingma e Welling (2014) risolve questo problema per distribuzioni per cui $z \sim Q(z; \lambda)$ può essere scritto come $z = g(\varepsilon, \lambda)$ dove $\varepsilon \sim p(\varepsilon)$ è una variabile di rumore con distribuzione fissa (indipendente da λ) e g è una funzione differenziabile. Per la gaussiana $Q(z; \mu, \sigma^2) = \mathcal{N}(\mu, \sigma^2)$, la riparametrizzazione è $z = \mu + \sigma \varepsilon$ con $\varepsilon \sim \mathcal{N}(0, 1)$.

Sotto questa riparametrizzazione:

$$\begin{aligned} \nabla_{\lambda} \mathcal{L}(\lambda) &= \nabla_{\lambda} E_{\{\varepsilon \sim p(\varepsilon)\}} [\log P(x, g(\varepsilon, \lambda)) - \log Q(g(\varepsilon, \lambda); \lambda)] = \\ &= E_{\{\varepsilon \sim p(\varepsilon)\}} [\nabla_{\lambda} (\log P(x, g(\varepsilon, \lambda)) - \log Q(g(\varepsilon, \lambda); \lambda))] \end{aligned}$$

Il gradiente passa sotto il segno di aspettazione perché $p(\varepsilon)$ è indipendente da λ : il risultato è una media di gradienti che può essere stimata con campioni Monte Carlo.

Questo produce lo **stimatore del gradiente della ELBO** per il Variational Autoencoder (VAE):

$$\nabla_{\lambda} \mathcal{L} \approx (1/L) \sum_{l=1}^L \nabla_{\lambda} [\log P(x | g(\varepsilon^l, \lambda)) - \log Q(g(\varepsilon^l, \lambda); \lambda) + \log P(g(\varepsilon^l, \lambda))]$$

con $\varepsilon^l \sim p(\varepsilon)$ campionati i.i.d. La varianza di questo stimatore è molto inferiore a quella dello stimatore REINFORCE (che non usa la riparametrizzazione), rendendo l'addestramento stabile.

Appendice Tecnica III

Algebra completa dei transformer e attenzione lineare

III.1 Il transformer come composizione di operatori lineari e nonlineari

L'architettura transformer, nella sua forma originale di Vaswani et al. (2017), è una composizione di blocchi che alternano due tipi di operatori: il meccanismo di multi-head attention (MHA), che realizza interazioni globali tra le posizioni della sequenza, e la rete feed-forward per posizione (FFN), che realizza trasformazioni locali non lineari. Ogni blocco è equipaggiato di connessioni residuali e normalizzazione dei layer (LayerNorm). L'intera architettura può essere espressa con notazione algebrica precisa che evidenzia le proprietà strutturali di ogni componente.

Sia $X \in \mathbb{R}^{n \times d}$ la matrice di input al blocco l , dove n è la lunghezza della sequenza e d è la dimensione del modello. L'output del blocco l è:

$$X' = \text{LayerNorm}(X + \text{MHA}(X))$$

$$X'' = \text{LayerNorm}(X' + \text{FFN}(X'))$$

La connessione residuale $X + \text{MHA}(X)$ garantisce che il gradiente non svanisca attraverso la profondità (cfr. Parte Terza, sezione III.3). LayerNorm normalizza ogni vettore di posizione indipendentemente:

$$\text{LayerNorm}(x) = \gamma \odot (x - \mu(x)) / \sqrt{(\sigma^2(x) + \epsilon)} + \beta$$

dove $\mu(x) = (1/d) \sum_i x_i$ è la media, $\sigma^2(x) = (1/d) \sum_i (x_i - \mu(x))^2$ è la varianza, $\gamma, \beta \in \mathbb{R}^d$ sono parametri appresi di scala e traslazione, e $\epsilon > 0$ è una costante di stabilità numerica (tipicamente 10^{-5}). La normalizzazione è applicata lungo la dimensione d per ogni posizione $i \in \{1, \dots, n\}$ indipendentemente.

La rete FFN è applicata posizione per posizione con pesi condivisi tra le posizioni:

$$\text{FFN}(x) = W_2 \cdot \text{ReLU}(W_1 \cdot x + b_1) + b_2$$

dove $W_1 \in \mathbb{R}^{d_{\text{fxd}}}$, $W_2 \in \mathbb{R}^{d_{\text{xdf}}}$, e $d^f = 4d$ è la dimensione interna tipica. La struttura posizione per posizione implica che FFN non produce interazioni tra posizioni diverse: tutta la comunicazione inter-posizionale è confinata in MHA .

III.2 La fattorizzazione dei parametri nell'attenzione multi-testa

Il meccanismo MHA con h teste può essere riscritto in forma matriciale unificata che chiarisce la struttura della fattorizzazione. Siano $W_Q, W_K \in \mathbb{R}^{d_{\text{xd}}}$ e $W_V, W_O \in \mathbb{R}^{d_{\text{xd}}}$ le matrici di proiezione totali (prima della suddivisione in teste).

Per la testa i , la proiezione è realizzata da sottomatrici:

$$W_Q^i = W_Q[i \cdot d : (i+1) \cdot d, :] \in \mathbb{R}^{d_{\text{kxd}}}$$

$$W_K^i = W_K[i \cdot d : (i+1) \cdot d, :] \in \mathbb{R}^{d \times d}$$

$$W_V^i = W_V[i \cdot d_v : (i+1) \cdot d_v, :] \in \mathbb{R}^{d_v \times d}$$

con $d = d_v = d/h$. L'output della testa i è:

$$\begin{aligned} \text{head}_i &= \text{Softmax}(X \cdot (W_Q^i)^T \cdot (X \cdot (W_K^i)^T)^T / \sqrt{d}) \cdot X \cdot (W_V^i)^T \\ &= \text{Softmax}(Q_i \cdot K_i^T / \sqrt{d}) \cdot V_i \end{aligned}$$

Il concatenamento e la proiezione finale:

$$\text{MHA}(X) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \cdot W_O^T$$

dove Concat impila gli h output di dimensione $n \times d_v$ in una matrice $n \times (h \cdot d_v) = n \times d$, e $W_O \in \mathbb{R}^{d \times d}$ riproietta nello spazio d -dimensionale.

La struttura di basso rango è il tratto algebrico saliente: ogni testa opera nello spazio delle coppie (Q, K) di dimensione $d = d/h \ll d$, fattorizzando la matrice di interazione completa $d \times d$ in h contributi di rango d . L'espressività del MHA è dunque quella di h diversi meccanismi di basso rango che operano in parallelo, la cui somma approssima una matrice di interazione di rango potenzialmente pieno.

III.3 L'algebra dell'attenzione: la matrice di Gram e le sue proprietà spettrali

La matrice centrale del meccanismo di attenzione è la matrice dei punteggi $A = Q \cdot K^T / \sqrt{d} \in \mathbb{R}^{n \times n}$, dopo l'applicazione della softmax per righe:

$$\tilde{A}[i,j] = \exp(A[i,j]) / \sum_k \exp(A[i,k])$$

$\tilde{A}$ è una matrice stocastica per righe: ogni riga somma a 1, ogni elemento è non-negativo. L'output dell'attenzione è $\tilde{A} \cdot V$, una media pesata delle righe di V .

Le proprietà spettrali di A determinano il comportamento del meccanismo. Si consideri il caso degenere $Q = K = X$ (self-attention senza proiezioni). Allora:

$$A = X \cdot X^T / \sqrt{d}$$

che è la matrice di Gram di X (a meno del fattore di scala). La matrice di Gram è semidefinita positiva con autovalori $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$. La struttura della softmax trasforma questi autovalori: per una matrice Gram con un autovalore dominante $\lambda_1 \gg \lambda_2$, la softmax concentra i pesi di attenzione sulle direzioni principali — il meccanismo seleziona le posizioni più "simili" alla query in modo sempre più netto all'aumentare di $\lambda_1/\sqrt{d}$.

Il fattore $\sqrt{d}$ ha dunque un'interpretazione spettrale precisa: normalizza la dispersione degli autovalori della matrice di Gram. Per componenti i.i.d. di media zero e varianza unitaria in Q e K , il raggio spettrale atteso di $Q \cdot K^T$ è $O(\sqrt{n \cdot d})$, e il fattore $1/\sqrt{d}$ lo riduce a $O(\sqrt{n})$, rendendo il comportamento della softmax indipendente da d (a meno di un fattore logaritmico nella dimensione tipica).

III.4 L'attenzione con maschera causale e i transformer autoregressivi

Nei modelli linguistici autoregressivi (GPT e varianti), l'attenzione è vincolata dal requisito di causalità: la posizione i non può "vedere" le posizioni $j > i$. Questo è implementato attraverso una maschera additiva $M \in \mathbb{R}^{n \times n}$:

$$A_{\text{masked}}[i,j] = A[i,j] + M[i,j]$$

dove $M[i,j] = 0$ se $j \leq i$ e $M[i,j] = -\infty$ se $j > i$. Applicando softmax a A_{masked} , i termini con $M[i,j] = -\infty$ producono $\exp(-\infty) = 0$, eliminando effettivamente le connessioni causalmente illecite.

La struttura triangolare inferiore della maschera impone alla matrice $\tilde{A}$ di essere strettamente triangolare inferiore (con diagonale non nulla).

L'output:

$$\text{Output}[i,:] = \sum_{j \leq i} \tilde{A}[i,j] \cdot V[j,:]$$

dipende soltanto dalle posizioni $j \leq i$: il modello al passo i ha accesso solo al contesto passato. Questa struttura è compatibile con la generazione autoregressiva token per token: al passo t , si estende la sequenza di un token calcolando solo la riga t della matrice di attenzione.

L'efficienza computazionale della generazione è migliorata dalla key-value cache (KV-cache): le coppie (K_j, V_j) per $j < t$ sono calcolate una sola volta e memorizzate. Al passo t , si calcola solo la nuova riga t di A , riducendo la complessità per passo da $O(t^2 \cdot d)$ a $O(t \cdot d)$ con accesso alla cache.

III.5 La complessità computazionale esatta e i colli di bottiglia

La complessità computazionale del transformer ha tre componenti principali.

Attenzione standard: Per una sequenza di lunghezza n e dimensione d :

- Proiezioni Q, K, V : $O(n \cdot d^2)$ per le tre matrici
- Matrice di punteggi $Q \cdot K^T$: $O(n^2 \cdot d)$ per testa, $O(n^2 \cdot d)$ totale per $h > \text{teste}$
- Softmax e aggregazione $\tilde{A} \cdot V$: $O(n^2)$ per testa, $O(n^2 \cdot h)$ totale

- Proiezione output W_O : $O(n \cdot d^2)$

Complessità totale per layer: $O(n^2 \cdot d + n \cdot d^2)$. Per $n \ll d$ (sequenze corte), domina il termine quadratico in d (le proiezioni). Per $n \gg d$ (sequenze lunghe), domina il termine quadratico in n (la matrice di attenzione).

FFN: $O(n \cdot d \cdot d^f) = O(n \cdot d^2)$ con $d^f = 4d$.

Memoria: La matrice di attenzione $\tilde{A}$ richiede $O(n^2 \cdot h)$ di memoria, che diventa il collo di bottiglia per n grande. Per GPT-4 con $n = 128000$ token, la matrice di attenzione (non compressa) richiederebbe $O(128000^2 \cdot h) \approx 10^{10}$ elementi per layer — computazionalmente proibitivo senza ottimizzazioni.

III.6 L'attenzione lineare: la riformulazione kernel

Il problema della complessità quadratica in n ha motivato lo sviluppo dell'attenzione lineare, introdotta da Katharopoulos et al. (2020) come approssimazione della softmax-attention attraverso una riformulazione kernel.

L'osservazione chiave è che la softmax può essere approssimata da un kernel positivo. Siano $\varphi: \mathbb{R}^{d_k} \rightarrow \mathbb{R}^r$ una funzione di feature mapping tale che:

$$\exp(q_i \cdot k / \sqrt{d}) \approx \varphi(q_i)^T \cdot \varphi(k)$$

Se questa approssimazione è valida, l'output dell'attenzione (senza normalizzazione) diventa:

$$\begin{aligned} \text{Output_unnorm}[i,:] &= \sum_j \varphi(q_i)^T \varphi(k_j) \cdot v^T \\ &= \varphi(q_i)^T \cdot (\sum_j \varphi(k_j) \cdot v^T) \\ &= \varphi(q_i)^T \cdot S \end{aligned}$$

dove $S = \sum \varphi(k) \cdot v^T \in \mathbb{R}^{r \times d_v}$ è una matrice di "contesto" che può essere calcolata una volta per tutte in $O(n \cdot r \cdot d_v)$. La normalizzazione è analoga:

$$z[i] = \sum \varphi(q_i)^T \varphi(k) = \varphi(q_i)^T \cdot (\sum \varphi(k)) = \varphi(q_i)^T \cdot s$$

dove $s = \sum \varphi(k) \in \mathbb{R}^r$. L'output normalizzato è:

$$\text{Output}[i,:] = (\varphi(q_i)^T \cdot S) / (\varphi(q_i)^T \cdot s)$$

La complessità totale è $O(n \cdot r \cdot (d + d_v))$, lineare in n . La chiave è che la sommatoria su j è computata una volta ($O(n \cdot r \cdot d_v)$) e poi condivisa tra tutte le query: non è necessario calcolare esplicitamente la matrice $n \times n$ dei punteggi.

III.7 Le scelte di feature mapping: ELU, random Fourier features, e FAVOR+

La scelta del feature mapping φ determina la qualità dell'approssimazione.

Feature ELU (Katharopoulos et al.):

$$\varphi(x)[i] = \text{ELU}(x_i) + 1 = \max(x_i, 0) + (1 - e^{(-\max(-x_i, 0))}) + 1$$

Questa scelta garantisce $\varphi(x) \geq 0$ component-wise (necessario per interpretare i pesi come probabilità), ma l'approssimazione di $\exp(q \cdot k / \sqrt{d})$ è locale e di qualità variabile.

Random Fourier Features (Rahimi e Recht, 2007): Per un kernel shift-invariante $k(x,y) = k(x-y)$, il teorema di Bochner garantisce che k è la trasformata di Fourier di una misura di probabilità:

$$k(x-y) = \int e^{i\omega \cdot (x-y)} p(\omega) d\omega \approx (1/r) \sum \cos(\omega \cdot x + b) \cos(\omega \cdot y + b)$$

dove $\omega \sim p(\omega)$ e $b \sim \text{Uniforme}[0, 2\pi]$. Questo produce un feature mapping di dimensione r con proprietà di approssimazione statisticamente controllate. La softmax non è shift-invariante, ma può essere convertita attraverso la trasformazione $\tilde{q} = q - \max(q)$ (per stabilità numerica).

FAVOR+ (Choromanski et al., 2021): FAVOR (Fast Attention Via positive Orthogonal Random features) usa feature random non-negative per approssimare la softmax-attention:

$$\varphi(x) = (1/\sqrt{r}) \cdot \exp(-\|x\|^2/2) \cdot [\exp(\omega_1 \cdot x), \dots, \exp(\omega_r \cdot x)]$$

dove $\omega_1, \dots, \omega_r$ sono vettori random estratti con caratteristiche di ortogonalità. Il risultato:

$$E[\varphi(q)^T \varphi(k)] = \exp(q \cdot k)$$

L'aspettazione è esatta per questo kernel: FAVOR+ è uno stimatore non-distorto della softmax-attention (con riscalatura appropriata). L'ortogonalità dei vettori random riduce la varianza dello stimatore rispetto al campionamento i.i.d.

III.8 I transformer ricorrenti: la struttura di stato nascosto dell'attenzione lineare

Una delle conseguenze algebriche più significative dell'attenzione lineare è che i transformer autoregressivi con attenzione lineare ammettono una formulazione ricorrente esatta, equivalente alle RNN ma con proprietà di addestramento superiori.

Nella forma causale (maschera triangolare inferiore), l'attenzione lineare all'istante t è:

$$\text{Output}[t,:] = \varphi(q_{-t})^T \cdot S / (\varphi(q_{-t})^T \cdot s)$$

dove gli stati di contesto accumulano le informazioni passate:

$$S = S_{-1} + \varphi(k) \cdot v^T \in \mathbb{R}^{r \times d_v}$$

$$s = s_{-1} + \varphi(k) \in \mathbb{R}^r$$

con $S_0 = 0, s_0 = 0$. La struttura è quella di una RNN con stato nascosto (S, s) di dimensione $r \times d_v + r = r(d_v + 1)$. L'aggiornamento è un'operazione di rango 1 (prodotto esterno $\varphi(k) \cdot v^T$) aggiunta allo stato. La query $\varphi(q)$ interroga lo stato tramite un prodotto matrice-vettore.

Questa equivalenza è profonda: dimostra che il transformer con attenzione lineare non è architetturealmente diverso da una RNN — è una RNN con una forma specifica dello stato nascosto (una matrice invece di un vettore) e una regola di aggiornamento di rango 1. La differenza computazionale è nel training: la struttura parallela dell'attenzione lineare permette addestramento efficiente su GPU attraverso il calcolo di S in modo cumulativo batch, mentre la struttura ricorrente rende l'inferenza $o(1)$ per passo (invece di $O(n)$ per l'attenzione standard con KV-cache).

III.9 Retentive Networks e la forma generalizzata

Il modello Retentive Network (RetNet) di Sun et al. (2023) generalizza l'attenzione lineare introducendo un fattore di decadimento $\gamma \in (0,1)$ nella ricorrenza:

$$S = \gamma \cdot S_{-1} + \varphi(k) \cdot v^T$$

Il fattore γ introduce un bias verso le informazioni recenti: le contribuzioni passate sono attenuate esponenzialmente come γ^{t-j} . Nella forma parallela (per addestramento):

$$\text{Output}[i,:] = \varphi(q_i)^T \cdot (\sum_j \gamma^{(i-j)} \varphi(k) \cdot v^T) / \text{normalizzazione}$$

L'output è equivalente a un'attenzione con pesi esponenzialmente decrescenti nel passato. La forma matriciale compatta usa una maschera di decadimento:

$$D[i,j] = \gamma^{(i-j)} \text{ per } j \leq i, 0 \text{ altrimenti}$$

$$\text{e l'output è } \tilde{A}_D \cdot V \text{ dove } \tilde{A}_D[i,j] \propto \varphi(q_i)^T \varphi(k) \cdot D[i,j].$$

La famiglia di modelli State Space Models (SSM), in particolare S4 di Gu et al. (2022) e Mamba di Gu e Dao (2023), generalizza ulteriormente questa struttura a transizioni di stato apprese:

$$h = A \cdot h_{-1} + B \cdot x$$

$$y = C \cdot h + D \cdot x$$

dove $A \in \mathbb{R}^{d \times d}$ è la matrice di transizione, $B \in \mathbb{R}^{d \times 1}$, $C \in \mathbb{R}^{1 \times d}$, $D \in \mathbb{R}$ (scalare). S4 parametrizza A come matrice normale diagonalizzabile con autovalori sull'asse immaginario, garantendo stabilità e permettendo il calcolo efficiente attraverso la convoluzione nella rappresentazione spettrale.

III.10 L'analisi in frequenza dell'attenzione: la prospettiva spettrale

Un approccio alternativo all'analisi del transformer usa la trasformata di Fourier discreta (DFT) per studiare le proprietà spettrali delle operazioni di attenzione.

Per una sequenza periodica $x: \{0, \dots, n-1\} \rightarrow \mathbb{R}$, la DFT è:

$$X[k] = \sum_{j=0}^{n-1} x[j] \cdot \exp(-2\pi i \cdot j \cdot k / n) \quad k = 0, \dots, n-1$$

$$\text{con inversa } x[j] = (1/n) \sum_k X[k] \cdot \exp(2\pi i \cdot j \cdot k / n).$$

La convoluzione circolare corrisponde al prodotto elemento per elemento in frequenza: $(x * y)[j] = \sum_k x[j-k \bmod n] y[k]$ si trasforma in $X \cdot Y$. Questo ha un'implicazione diretta per l'attenzione: se i pesi $\tilde{A}[i,j]$ dipendono solo dalla differenza $i-j$ (shift-invariant attention), allora l'attenzione è una convoluzione e può essere calcolata in $O(n \log n)$ attraverso la FFT.

I transformer con Rotary Position Embedding (RoPE, Su et al. 2022) hanno parzialmente questa struttura. RoPE parametrizza le posizioni attraverso rotazioni nel piano complesso:

$$\tilde{q}[i, 2l-1:2l] = [q[i, 2l-1] \cos(\theta_i) - q[i, 2l] \sin(\theta_i),$$

$$q[i, 2l-1] \sin(\theta_i) + q[i, 2l] \cos(\theta_i)]$$

dove $\theta^l = i / 10000^{(2l/d)}$. Il prodotto interno $\tilde{q}_i \cdot \tilde{k}_j$ dipende solo dalla differenza di posizione $i-j$ attraverso termini del tipo $\cos((i-j)\theta^l)$, rendendo il meccanismo invariante alle traslazioni della sequenza e permettendo un'efficiente gestione delle sequenze lunghe attraverso interpolazione in frequenza.

III.11 La normalizzazione dei layer e la sua giustificazione variazionale

La scelta di LayerNorm rispetto a BatchNorm nel transformer non è arbitraria: ha una giustificazione variazionale che connette l'architettura alla geometria dell'informazione sviluppata nell'Appendice II.

BatchNorm (Ioffe e Szegedy, 2015) normalizza su elementi dello stesso minibatch:

$$\text{BN}(x_i) = (x_i - \mu_B) / \sqrt{(\sigma^2_B + \epsilon)} \cdot \gamma + \beta$$

dove μ_B , σ^2_B sono calcolati sull'intero minibatch. BatchNorm presuppone i.i.d. degli elementi del minibatch e ha comportamento degradato per batch piccoli e sequenze variabili.

LayerNorm (Ba et al., 2016) normalizza sulle feature di ogni singolo elemento:

$$\text{LN}(x) = (x - \mu(x)) / \sqrt{(\sigma^2(x) + \epsilon)} \cdot \gamma + \beta$$

dove $\mu(x) = (1/d) \sum_i x_i$. LayerNorm è invariante alle permutazioni dei campioni nel batch e alle trasformazioni di scala dell'input — proprietà fondamentale per il processing delle sequenze dove gli elementi hanno diverse scale caratteristiche.

La giustificazione variazionale: LayerNorm proietta ogni rappresentazione x sulla sfera unitaria (a meno delle riscature γ , β). Questa proiezione è analoga alla normalizzazione che porta dalla metrica euclidea alla metrica di Fisher nella geometria dell'informazione: normalizza le rappresentazioni in modo da rendere comparabili vettori in scale diverse. Per le distribuzioni gaussiane, la metrica di Fisher sulla sfera di Fisher coincide con la metrica di riemanniana sferica, e LayerNorm è la proiezione naturale su questa geometria.

RMSNorm (Zhang e Sennrich, 2019): semplificazione che normalizza solo la norma RMS senza centrare:

$$\text{RMSNorm}(x) = x / \text{RMS}(x) \cdot \gamma \text{ dove } \text{RMS}(x) = \sqrt{(1/d) \sum_i x_i^2}$$

Teoricamente più semplice di LayerNorm (rimuove il termine di centratura), RMSNorm è la scelta di normalizzazione in LLaMA e modelli successivi, con performances empiriche equivalenti e costo computazionale inferiore.

III.12 La decomposizione SVD dei pesi e l'approssimazione di basso rango

Un'analisi algebrica fondamentale dei transformer riguarda la struttura di rango delle matrici di pesi apprese. Sia $W \in \mathbb{R}^{m \times n}$ una matrice di pesi generica. La sua decomposizione a valori singolari (SVD) è:

$$W = U \cdot \Sigma \cdot V^T$$

dove $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ sono matrici ortogonali e $\Sigma \in \mathbb{R}^{m \times n}$ è diagonale con $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$ (valori singolari).

La tecnica LoRA (Low-Rank Adaptation, Hu et al. 2022) si basa sull'osservazione che le modifiche alle matrici di pesi durante il fine-tuning hanno basso rango intrinseco. LoRA parametrizza l'aggiornamento come:

$$\Delta W = A \cdot B \text{ con } A \in \mathbb{R}^{m \times r}, B \in \mathbb{R}^{r \times n}, r \ll \min(m, n)$$

La matrice aggiornata è $W_0 + \Delta W = W_0 + A \cdot B$. Poiché $\text{rank}(A \cdot B) \leq r$, l'aggiornamento è di rango al più r . Il numero di parametri trainabili si riduce da $m \cdot n$ a $r \cdot (m + n)$, un risparmio del fattore $mn / (r(m + n)) \approx \min(m, n) / (2r)$ per matrici quadrate.

La giustificazione teorica di LoRA è connessa al fenomeno della doppia discesa (Parte B): nel regime di sovrapparametrizzazione, i pesi del modello base W_0 occupano già una regione di alta norma nello spazio dei parametri. Le perturbazioni utili per adattare il modello a nuovi compiti sono localizzate in sottospazi di bassa dimensione — i sottospazi corrispondenti ai grandi valori singolari dell'aggiornamento.

La stima empirica del rango intrinseco (Aghajanyan et al. 2020) ha mostrato che la maggior parte dei compiti di fine-tuning richiede

modifiche in sottospazi di dimensione $r \leq 8$ su matrici di dimensione $d \geq 768$, confermando l'ipotesi di LoRA su basi empiriche.

III.13 La proiezione nell'attenzione e il teorema di approssimazione universale

Il transformer con MHA e FFN non-lineare è un approssimatore universale nel senso seguente. Il teorema di approssimazione universale per transformer (Yun et al. 2020) afferma:

Per ogni funzione continua $f: [0,1]^{n \times d} \rightarrow [0,1]^{n \times d}$ (su sequenze normalizzate) e ogni $\epsilon > 0$, esiste un transformer con un numero finito di layer, teste di attenzione, e dimensione FFN, tale che $\|T(X) - f(X)\| < \epsilon$ per ogni $X \in [0,1]^{n \times d}$.

La dimostrazione è costruttiva in due passi:

1. Si mostra che il meccanismo di attenzione softmax con proiezioni > trainabili può approssimare arbitrariamente bene qualunque > funzione continua sulle coppie (posizione, valore), incluse le > funzioni che implementano ordinamenti e selezioni.
2. Si mostra che la FFN con ReLU è un approssimatore universale per > funzioni scalari (teorema di Cybenko 1989, nella versione per reti > a larghezza arbitraria), e la composizione con l'attenzione > estende l'universalità all'intero transformer.

Il teorema è esistenziale e non costruttivo nel senso che non fornisce la profondità o la larghezza minima per approssimare una funzione data. I bound costruttivi richiedono $O(1/\epsilon^2)$ parametri per l'approssimazione di funzioni Lipschitz in dimensione d , con dipendenza esponenziale in d per funzioni generali (curse of dimensionality).

Nota bibliografica dell'Appendice III

Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser e Polosukhin, Attention Is All You Need, in Proceedings of NeurIPS (2017). Ba, Kiros e Hinton, Layer Normalization, arXiv:1607.06450 (2016). Katharopoulos, Vyas, Pappas e Fleuret, Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention, in Proceedings of ICML (2020). Choromanski, Likhoshesterov, Dohan, Song, Gane, Sarlos, Hawkins, Davis, Mohiuddin, Kaiser, Belanger, Colwell e Weller, Rethinking Attention with Performers, in Proceedings of ICLR (2021). Su, Lu, Pan, Murtadha, Wen e Liu, RoFormer: Enhanced Transformer with Rotary Position Embedding, arXiv:2104.09864 (2022). Sun, Dong, Huang, Ma, Patra, Sinha, Chaudhry, Nikan, Zhou, Wu e Wei, Retentive Network: A Successor to Transformer for Large Language Models, arXiv:2307.08621 (2023). Gu, Goel e Ré, Efficiently Modeling Long Sequences with Structured State Spaces, in Proceedings of ICLR (2022). Gu e Dao, Mamba: Linear-Time Sequence Modeling with Selective State Spaces, arXiv:2312.00752 (2023). Hu, Shen, Wallis, Allen-Zhu, Li, Wang, Wang e Chen, LoRA: Low-Rank Adaptation of Large Language Models, in Proceedings of ICLR (2022). Yun, Bhojanapalli, Rawat, Reddi e Kumar, Are Transformers Universal Approximators of Sequence-to-Sequence Functions?, in Proceedings of ICLR (2020). Aghajanyan, Zettlemoyer e Gupta, Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning, in Proceedings of ACL (2021). Zhang e Sennrich, Root Mean Square Layer Normalization, in Proceedings of NeurIPS (2019). Rahimi e Recht, Random Features for Large-Scale Kernel Machines, in Proceedings of NeurIPS (2007).

Appendice Tecnica IV

Teoria delle categorie applicata ai sistemi formali e alle reti neurali: funtori, aggiunzioni di Kan, monadi e architettura dei modelli

IV.1 Categorie, funtori e trasformazioni naturali: il linguaggio

La teoria delle categorie — sviluppata da Eilenberg e Mac Lane nel 1945 come strumento per formalizzare le "naturali" trasformazioni tra costruzioni matematiche — è diventata il linguaggio unificante della matematica strutturale e, più recentemente, un framework per ragionare sulle architetture dei sistemi formali e computazionali. L'obiettivo di questa appendice è costruire il ponte tra le strutture categoriali astratte e le concrete architetture dei sistemi descritti nelle parti principali della monografia.

Una **categoria** $\mathcal{C}$ consiste di:

- una collezione $\text{Ob}(\mathcal{C})$ di **oggetti** $A, B, C, \dots$
- per ogni coppia di oggetti A, B , un insieme $\text{Hom}(A, B)$ di **morfismi** $f: A \rightarrow B$
- per ogni oggetto A , un morfismo identità $\text{id}_A: A \rightarrow A$
- un'operazione di **composizione**: se $f: A \rightarrow B$ e $g: B \rightarrow C$, allora $g \circ f: A \rightarrow C$

soddisfacenti gli assiomi:

(i) Associatività: $h \circ (g \circ f) = (h \circ g) \circ f$

(ii) Identità: $\text{id}_B \circ f = f = f \circ \text{id}_A$

Le categorie fondamentali per i nostri scopi:

Set: oggetti = insiemi, morfismi = funzioni totali. La composizione è la composizione di funzioni.

Vect_k: oggetti = spazi vettoriali su un campo k , morfismi = trasformazioni lineari. L'identità è l'identità lineare. Questa è la categoria in cui vivono le reti neurali (i layer sono morfismi tra spazi vettoriali).

Top: oggetti = spazi topologici, morfismi = funzioni continue. La topologia dello spazio dei parametri (cfr. Parte Terza, sezione III.7) è studiata in Top.

Pos: oggetti = insiemi parzialmente ordinati (poset), morfismi = funzioni monotone. Questa è la categoria naturale per la teoria degli ordini (Appendice VI).

Cat: la categoria delle categorie stesse, con oggetti = categorie e morfismi = funtori. Questa è la struttura metacategoriale.

IV.2 Funtori e la struttura dei modelli formali

Un **funtore** $F: \mathcal{C} \rightarrow \mathcal{D}$ è una mappa che preserva la struttura categoriale:

$$F: \text{Ob}(\mathcal{C}) \rightarrow \text{Ob}(\mathcal{D})$$

$$F: \text{Hom}_{\mathcal{C}}(A, B) \rightarrow \text{Hom}_{\mathcal{D}}(F(A), F(B))$$

soddisfacendo:

$$F(\text{id}_A) = \text{id}_{\{F(A)\}}$$

$$F(g \circ f) = F(g) \circ F(f)$$

I funtori covarianti preservano la direzione dei morfismi; i **funtori controvarianti** la invertono: $F(f: A \rightarrow B) = F(f): F(B) \rightarrow F(A)$.

Esempio fondamentale: il **funtore di dimenticanza** (forgetful functor) $U: \mathbf{Grp} \rightarrow \mathbf{Set}$ manda ogni gruppo G al suo insieme sottostante $U(G)$, dimenticando la struttura di gruppo. Ogni omomorfismo di gruppi $\varphi: G \rightarrow H$ diventa una funzione di insiemi $U(\varphi): U(G) \rightarrow U(H)$. Il funtore U non è invertibile: non si può "ricordare" la struttura dimenticata in modo canonico.

Il collegamento con i sistemi formali: il processo di interpretazione di un sistema formale (la funzione semantica $\llbracket \cdot \rrbracket$ che manda sintassi in semantica) è un funtore. Più precisamente, sia **Syn** la categoria con oggetti = formule e morfismi = derivazioni, e **Sem** la categoria con oggetti = modelli e morfismi = omomorfismi di modelli. La semantica denotazionale è un funtore $\llbracket \cdot \rrbracket: \mathbf{Syn} \rightarrow \mathbf{Sem}$.

Il teorema di completezza di Gödel per la logica del primo ordine afferma che questo funtore è "pieno" nel senso che ogni morfismo in **Sem** si solleva a **Syn**: se $\llbracket \varphi \rrbracket \leq \llbracket \psi \rrbracket$ in ogni modello, allora $\varphi \vdash \psi$ è derivabile. Il teorema di incompletezza afferma che per la semantica aritmetica standard ($\mathbb{N}$ come modello), il funtore non è fedele: ci sono morfismi in **Sem** (le verità aritmetiche) che non hanno preimmagine in **Syn** (non sono dimostrabili).

IV.3 Trasformazioni naturali e il concetto di naturalità

Una **trasformazione naturale** $\eta: F \Rightarrow G$ tra funtori $F, G: \mathcal{C} \rightarrow \mathcal{D}$ è una famiglia di morfismi $\{\eta_A: F(A) \rightarrow G(A)\}_{A \in \mathcal{C}}$ tale che per ogni $f: A \rightarrow B$ in $\mathcal{C}$ il **diagramma di naturalità** commuta:

$$\begin{array}{ccc} F(A) & \xrightarrow{\eta_A} & G(A) \\ | & & | \\ F(f) & & G(f) \end{array}$$

| |

v v

$$F(B) \xrightarrow{\eta_B} G(B)$$

$$\text{ossia } G(f) \circ \eta_A = \eta_B \circ F(f).$$

Le trasformazioni naturali catturano la nozione di "costruzione naturale" che Eilenberg e Mac Lane volevano formalizzare: una costruzione che non dipende da scelte arbitrarie. Il determinante di una matrice è una trasformazione naturale dal funtore $GL(n, -): \mathbf{CRing} \rightarrow \mathbf{Group}$ (il gruppo lineare generale sopra un anello commutativo) al funtore $(-)\times: \mathbf{CRing} \rightarrow \mathbf{Group}$ delle unità moltiplicative. La naturalità cattura il fatto che $\det$ non dipende dalla base scelta.

Nel contesto delle reti neurali, le trasformazioni naturali appaiono nella teoria dell'equivarianza. Un layer equivariante rispetto a un gruppo G è una mappa $f: V \rightarrow W$ tale che:

$$f(\varrho_V(g) \cdot v) = \varrho_W(g) \cdot f(v) \text{ per ogni } g \in G, v \in V$$

dove ϱ_V, ϱ_W sono le rappresentazioni di G su V e W . Questa è precisamente la condizione di naturalità per le trasformazioni naturali tra funtori $\varrho_V, \varrho_W: G \rightarrow \mathbf{Vect}_k$ (dove G è vista come categoria con un oggetto e morfismi = elementi del gruppo).

Le reti neurali equivarianti (Equivariant Neural Networks, Cohen e Welling 2016) sfruttano la struttura di gruppo per costruire architetture che rispettano le simmetrie dei dati: le CNN convoluzionali sono equivarianti rispetto alle traslazioni, le reti sferiche rispetto alle rotazioni $SO(3)$, i modelli molecolari rispetto al gruppo di simmetria delle molecole $E(3)$.

IV.4 Categorie chiuse cartesiane e il λ -calcolo

Le **categorie chiuse cartesiane** (CCC) sono categorie con prodotti e oggetti di hom-interni che forniscono il modello semantico del λ -calcolo semplicemente tipizzato — la connessione categoriale della corrispondenza di Curry-Howard discussa nella sezione D.6 della monografia.

Una categoria $\mathcal{C}$ è **cartesiana** se ha:

- un **oggetto terminale** 1 (tale che per ogni A esiste un unico $>$ morfismo $A \rightarrow 1$)
- **prodotti binari** $A \times B$ con proiezioni $\pi_1: A \times B \rightarrow A$, $\pi_2: A \times B \rightarrow B$ e $>$ la proprietà universale: per ogni $f: C \rightarrow A$, $g: C \rightarrow B$, esiste un $>$ unico $\langle f, g \rangle: C \rightarrow A \times B$

È **chiusa** se per ogni coppia A, B esiste un **oggetto esponenziale** B^A (o $A \Rightarrow B$) con una valutazione:

$\text{eval}: B^A \times A \rightarrow B$

e la proprietà universale: per ogni $f: C \times A \rightarrow B$, esiste un unico $\hat{f}: C \rightarrow B^A$ (la **currying** di f) tale che $\text{eval} \circ (\hat{f} \times \text{id}_A) = f$.

La corrispondenza con il λ -calcolo:

- oggetti = tipi
- morfismi $A \rightarrow B$ = termini ben tipizzati $x:A \vdash t:B$ (termini con $>$ variabile libera x di tipo A)
- prodotti = tipi prodotto $A \times B$ (coppie)
- esponenziali = tipi funzione $A \rightarrow B$
- composizione = sostituzione di termini

- currying = λ -astrazione: $f: C \times A \rightarrow B$ corrisponde a $\lambda x.f(-,x): C \rightarrow (A \rightarrow B)$

Questo è il contenuto della corrispondenza di Curry-Howard in versione categoriale: i tipi sono oggetti in una CCC, i termini sono morfismi, le prove sono programmi.

Set è una CCC: il prodotto è il prodotto cartesiano, l'oggetto esponenziale B^A è l'insieme di tutte le funzioni da A a B , $\text{eval}(f,a) = f(a)$. **Vect_k** non è una CCC nel senso stretto (il prodotto tensore non soddisfa la proprietà della CCC), ma la chiusura rispetto al prodotto tensore $\otimes$ produce la struttura di **categoria monoidale chiusa** — la struttura giusta per i tipi lineari.

IV.5 Monadi: il pattern computazionale degli effetti

Le **monadi** sono uno dei contributi più influenti della teoria delle categorie alla teoria della programmazione, introdotte formalmente da Eilenberg e Mac Lane (1945) e portate nell'informatica teorica da Moggi (1991) come modello semantico degli effetti computazionali. Una monade su una categoria $\mathcal{C}$ è una tripletta (T, η, μ) dove:

- $T: \mathcal{C} \rightarrow \mathcal{C}$ è un funtore endomorfo
- $\eta: \text{Id}_{\mathcal{C}} \Rightarrow T$ è la **unità** (o return), una trasformazione naturale $>$ dall'identità a T
- $\mu: T^2 \Rightarrow T$ è la **moltiplicazione** (o join), una trasformazione $>$ naturale da $T \circ T$ a T

soddisfacenti le leggi monadali:

$$\mu_A \circ T(\mu_A) = \mu_A \circ \mu_{\{T(A)\}} \text{ (associatività)}$$

$$\mu_A \circ T(\eta_A) = \text{id}_{\{T(A)\}} = \mu_A \circ \eta_{\{T(A)\}} \text{ (identità)}$$

Le monadi appaiono ovunque come pattern di "computazione con contesto":

La monade Maybe (Option): $T(A) = A \cup \{\text{Nothing}\}$, modella computazioni che possono fallire. $\eta_A(a) = \text{Just}(a)$, $\mu(\text{Just}(\text{Just}(a))) = \text{Just}(a)$, $\mu(\text{Just}(\text{Nothing})) = \text{Nothing}$.

La monade List: $T(A) = \text{liste finite di } A$, modella computazioni non-deterministiche. $\eta_A(a) = [a]$, $\mu = \text{flatten}$ (concatenazione di liste di liste).

La monade State S: $T(A) = S \rightarrow (A \times S)$, modella computazioni con stato mutabile. $\eta_A(a) = \lambda s.(a, s)$, $\mu = \text{composizione delle transizioni di stato}$.

La monade di distribuzione di probabilità: $T(A) = \text{distribuzioni di probabilità su } A$, modella computazioni probabilistiche. $\eta_A(a) = \text{distribuzione di Dirac } \delta_a$, $\mu = \text{operazione di marginalizzazione (integrazione)}$. Questa è la monade naturale per i modelli probabilistici — inclusi i modelli linguistici, dove $T(\text{Voc})$ è la distribuzione sul vocabolario.

IV.6 La monade di continuazione e il controllo computazionale

La **monade di continuazione** ha un'importanza speciale per la teoria della backpropagation e del gradiente automatico. Il tipo $T(A) = (A \rightarrow R) \rightarrow R$ (funzioni da "continuazioni" di tipo $A \rightarrow R$ al tipo di risposta R) cattura computazioni che hanno accesso esplicito alla propria continuazione.

La connessione con la backpropagation (Pearlmutter e Siskind 2008): la forward pass di una rete neurale è una computazione che produce un

valore e simultaneamente produce una continuazione che, se eseguita con un gradiente upstream, propaga il gradiente backwards. Formalmente, la derivata di una funzione $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ può essere rappresentata come:

$$D_f: \mathbb{R}^n \rightarrow (\mathbb{R}^n \rightarrow \mathbb{R}) \times (\mathbb{R}^m \rightarrow \mathbb{R})$$

$$D_f(x) = (f(x), \lambda \tilde{y}. J^f(x)^T \tilde{y})$$

dove $J^f(x)$ è la matrice Jacobiana di f in x . La prima componente è il valore forward, la seconda è la funzione che mappa cotangenti (gradienti upstream) a cotangenti (gradienti downstream). Questa struttura è precisamente quella della monade di continuazione con $R = \mathbb{R}$.

La trasformazione da "stile diretto" a "stile di continuazione" (CPS transformation) è un isomorfismo di categorie tra la categoria dei programmi in stile diretto e la categoria dei programmi in CPS. La backpropagation è la CPS transformation della forward pass — un fatto algebrico profondo che spiega perché la backpropagation è un algoritmo efficientemente derivabile dalla struttura della computazione in avanti, senza richiedere conoscenza della struttura specifica della rete.

IV.7 Aggiunzioni: il concetto più profondo della teoria delle categorie

La nozione di aggiunzione (adjunction) è, secondo Mac Lane, la "nozione fondamentale di tutta la matematica" — un'affermazione che la sua storia conferma ampiamente. Una aggiunzione tra funtori $F: \mathcal{C} \rightarrow \mathcal{D}$ (funtoe sinistro) e $G: \mathcal{D} \rightarrow \mathcal{C}$ (funtoe destro) è una biiezione naturale:

$$\text{Hom}_{\mathcal{D}}(F(A), B) \cong \text{Hom}_{\mathcal{C}}(A, G(B))$$

naturale in $A \in \mathcal{C}$ e $B \in \mathcal{D}$. Si scrive $F \dashv G$ (F è aggiunto sinistro di G , G è aggiunto destro di F).

Le aggiunzioni catturano le ottimizzazioni universali: $F(A)$ è "il miglior oggetto in $\mathcal{D}$ costruito a partire da A rispetto alla struttura di G ".
Equivalentemente:

Unità: $\eta: \text{Id}_{\mathcal{C}} \Rightarrow G \circ F$, con $\eta_A: A \rightarrow G(F(A))$ **Counità:** $\varepsilon: F \circ G \Rightarrow \text{Id}_{\mathcal{D}}$, con $\varepsilon_B: F(G(B)) \rightarrow B$

soddisfacendo le identità triangolari:

$$(\varepsilon_{F(A)}) \circ F(\eta_A) = \text{id}_{\{F(A)\}}$$

$$G(\varepsilon_B) \circ \eta_{\{G(B)\}} = \text{id}_{\{G(B)\}}$$

Esempi fondamentali che riappaiono nella monografia:

Prodotto $\dashv$ Esponenziale (currying-uncurrying):

$$\text{Hom}(A \times B, C) \cong \text{Hom}(A, C^B)$$

L'aggiunzione currying-uncurrying è il fondamento algebrico della λ -astrazione.

Forgetful $\dashv$ Free (dimenticanza-libertà): Il funtore di dimenticanza $U: \text{Grp} \rightarrow \text{Set}$ ha un aggiunto sinistro $F: \text{Set} \rightarrow \text{Grp}$ che manda ogni insieme X nel gruppo libero $F(X)$ generato da X . L'aggiunzione:

$$\text{Hom}_{\text{Grp}}(F(X), G) \cong \text{Hom}_{\text{Set}}(X, U(G))$$

dice che i morfismi di gruppi da $F(X)$ a G sono in biiezione con le funzioni da X a $U(G)$: un omomorfismo di gruppi dal gruppo libero è determinato dai valori sulle generatrici.

Quantificatori come aggiunzioni: In logica del primo ordine, il quantificatore esistenziale $\exists_x$ è l'aggiunto sinistro della sostituzione $\Sigma_{\{x:A\}}$, e il quantificatore universale $\forall_x$ è l'aggiunto destro:

$$\Sigma_{\{x:A\}} \dashv x^* \dashv \Pi_{\{x:A\}}$$

dove $x^*: \text{Prop}(\Gamma) \rightarrow \text{Prop}(\Gamma, x:A)$ è la "debolezza" che aggiunge la variabile x al contesto. Questa è la formulazione categoriale della semantica dei quantificatori.

IV.8 Aggiunzioni di Kan: l'estensione universale dei funtori

Le **aggiunzioni di Kan** (Kan extensions), introdotte da Daniel Kan nel 1958, sono il modo categoriale di estendere un funtore lungo un'inclusione. Sono, secondo Mac Lane, il concetto "ubiquitario" della teoria delle categorie.

Siano $K: \mathcal{A} \rightarrow \mathcal{B}$ e $F: \mathcal{A} \rightarrow \mathcal{C}$ funtori. La **estensione di Kan sinistra** (Left Kan Extension) di F lungo K è un funtore $\text{Lan}_K F: \mathcal{B} \rightarrow \mathcal{C}$ insieme a una trasformazione naturale $\eta: F \Rightarrow (\text{Lan}_K F) \circ K$, **universale** nel senso:

per ogni funtore $G: \mathcal{B} \rightarrow \mathcal{C}$ e trasformazione naturale $\alpha: F \Rightarrow G \circ K$, esiste una **unica** trasformazione naturale $\tilde{\alpha}: \text{Lan}_K F \Rightarrow G$ tale che $\alpha = (\tilde{\alpha} K) \circ \eta$:

$$\mathcal{A} \xrightarrow{K} \mathcal{B}$$

$\backslash /$

$$F \text{ Lan}_K F$$

$\backslash /$

$$\mathcal{C}$$

La formula coend per $\text{Lan}_K F$ quando $\mathcal{C}$ ha colimiti sufficienti:

$$(\text{Lan}_K F)(B) = \int^{\wedge} \{A \in \mathcal{A} \mid \text{Hom}_{\mathcal{B}}(K(A), B) \otimes F(A)$$

dove $\int^{\wedge} A$ denota il **coend** — il colimite "torto" che generalizza la coend della teoria degli insiemi ($\int = \sum$ nel caso discreto).

Connessione con l'apprendimento automatico: Le Kan extensions appaiono nella teoria della generalizzazione del machine learning come il framework universale per l'interpolazione e l'estrapolazione dei dati. Se $F: \mathcal{D}_{\text{train}} \rightarrow \mathbb{R}$ è la funzione di addestramento (l'input-output mapping sui dati di training) e $K: \mathcal{D}_{\text{train}} \hookrightarrow \mathcal{X}$ è l'inclusione dei dati di training nell'intero spazio degli input, allora:

- la Left Kan Extension $\text{Lan}_K F$ generalizza F all'intera $\mathcal{X}$ nel modo > "più libero possibile" (zero assunzioni extra)
- la Right Kan Extension $\text{Ran}_K F$ generalizza nel modo "più vincolato > possibile"

Il modello appreso da una rete neurale è, in questo senso, un'approssimazione della Left o Right Kan Extension a seconda della struttura della funzione di perdita e della regolarizzazione: la perdita quadratica con regolarizzazione L2 approssima la Ran (regolarizzazione verso il default), mentre la perdita entropica approssima la Lan .

Fong e Spivak (2019) hanno sviluppato questo collegamento sistematicamente nel contesto delle **Para-categorie** e dell'apprendimento supervisionato categoriale.

IV.9 Reti neurali come morfismi in una categoria di mappe parametrizzate

Il framework di Fong, Spivak e Tuyeras (2019) per le reti neurali come sistemi compositivi usa la struttura di **categoria moroidale** per formalizzare la composizione dei layer.

Una **categoria moroidale** $(\mathcal{C}, \otimes, I, \alpha, \lambda, \varrho)$ è una categoria con:

- un **funtore prodotto** $\otimes: \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$

- un **oggetto unità** I
- isomorfismi naturali di associatività α , unità sinistra λ , unità > destra ϱ

soddisfacenti le equazioni coerenza (diagramma del pentagono per α , diagramma del triangolo per λ, ϱ).

La categoria **Para**($\mathcal{C}$) delle mappe parametrizzate su $\mathcal{C}$:

- oggetti = oggetti di $\mathcal{C}$
- morfismi $A \rightarrow B$ in $\text{Para}(\mathcal{C})$ = coppie (P, f) dove $P \in \text{Ob}(\mathcal{C})$ è lo spazio > dei parametri e $f: P \otimes A \rightarrow B$ è la funzione parametrizzata
- composizione: $(P, f) \circ (Q, g) = (P \otimes Q, h)$ dove $h(p, q, x) = f(p, > g(q, x))$

Un layer di rete neurale con pesi W e funzione di attivazione σ è un morfismo in $\text{Para}(\text{Vect}_{\mathbb{R}})$:

$$(\mathbb{R}^{(m \times n)}, f_W): \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ dove } f_W(W, x) = \sigma(W \cdot x + b)$$

La composizione in $\text{Para}(\text{Vect}_{\mathbb{R}})$ è esattamente la composizione dei layer:

$$(\mathbb{R}^{(m \times n)}, f_{W_1}) \circ (\mathbb{R}^{(n \times k)}, f_{W_2}) = (\mathbb{R}^{(m \times n)} \times \mathbb{R}^{(n \times k)}, f_{\{W_1, W_2\}})$$

La backpropagation è un funtore da $\text{Para}(\text{Vect}_{\mathbb{R}})$ a una categoria di "lenti" (optics) che accoppia la forward pass con la backward pass:

$$\text{Backprop}: \text{Para}(\text{Vect}_{\mathbb{R}}) \rightarrow \text{Lens}(\text{Vect}_{\mathbb{R}})$$

dove una lente (L, F) consiste di una funzione di osservazione $F: L \times S \rightarrow V$ (forward) e una funzione di aggiornamento $F^\dagger: L \times S \times V \rightarrow L \times$

S (backward). La lente per un layer è esattamente la coppia (forward_pass, backward_pass).

IV.10 Topos e la geometria dei sistemi logici

Un **topos** (plurale: topoi o toposes) è una categoria che si comporta come la categoria degli insiemi ma può avere geometria interna non-triviale. La definizione formale: un topos è una categoria $\mathcal{C}$ che:

1. ha tutti i limiti finiti (prodotti finiti, equalizzatori)
2. è cartesianamente chiusa (ha esponenziali A^B per ogni A, B)

ha un **classificatore di sottooggetti** Ω : un oggetto con un morfismo "vero" $\top: 1 \rightarrow \Omega$ tale che per ogni sottooggetto $m: A' \hookrightarrow A$ esiste un unico $\chi_m: A \rightarrow \Omega$ (il "caratteristico" di m) che fa commutare:

$$A' \xrightarrow{\quad} 1 \mid \mid m \mid \mid \mid \vee \vee A \xrightarrow{\quad} \chi_m \xrightarrow{\quad} \Omega$$

In **Set**, $\Omega = \{0, 1\} = \mathbb{B}$ (i booleani) e $\chi_m(a) = 1 \leftrightarrow a \in A'$. In un topos generale, Ω non è booleano: i "valori di verità" possono formare una struttura algebrica complessa (un'algebra di Heyting — cfr. Appendice VI).

Il topos dei fasci $\text{Sh}(X)$ su uno spazio topologico X ha:

- oggetti = fasci su X (assegnazioni localmente coerenti di insiemi > agli aperti di X)
- Ω = il fascio degli aperti: $\Omega(U) = \{V \text{ aperto} : V \subseteq U\}$

I valori di verità sono aperti di X — non solo "vero/falso" ma "vero nell'aperto U ". Questo modella la logica temporale e spaziale: una proposizione è vera in un contesto (aperto) se è vera localmente.

Connessione con i sistemi formali: ogni topos è dotato di una logica interna — la logica intuizionista del topos — in cui le formule del primo

ordine possono essere interpretate come morfismi nell'oggetto Ω . La logica interna di Set è la logica classica (Ω booleano); la logica interna di $\text{Sh}(X)$ è intuizionista con valori di verità topologici. Il teorema di completezza per la logica del primo ordine (in un'opportuna categoria di topos di insiemi) è una manifestazione della corrispondenza tra la sintassi logica e la struttura del classificatore Ω .

IV.11 Categorie arricchite e la struttura metrica delle reti neurali

Le **categorie arricchite** generalizzano le categorie ordinarie sostituendo i set di morfismi $\text{Hom}(A,B)$ con oggetti di una categoria moroidale $\mathcal{V}$. Se $\mathcal{V} = \text{Set}$, si ottiene le categorie ordinarie. Se $\mathcal{V} = ([0,\infty], \geq, +, 0)$ — l'intervallo non-negativo esteso con ordine inverso, somma come prodotto moroidale e 0 come unità — si ottengono le **categorie arricchite su $[0,\infty]$** , che sono equivalenti agli **spazi metrici** (Lawvere 1973).

Un **spazio metrico generalizzato** (non necessariamente simmetrico) è una categoria arricchita su $([0,\infty], \geq)$: gli oggetti sono i punti, $\text{Hom}(A,B) = d(A,B)$ è la distanza, la legge di composizione è la disuguaglianza triangolare:

$$d(A,C) \leq d(A,B) + d(B,C)$$

e l'identità è $d(A,A) \leq 0$ (che con $d \geq 0$ dà $d(A,A) = 0$).

Questo collegamento ha implicazioni per le reti neurali: lo spazio delle funzioni apprese da una rete neurale è uno spazio metrico nella categoria arricchita appropriata, e le proprietà di continuità della funzione appresa corrispondono alle proprietà di arricchimento. La Lipschitz constraint nelle reti neurali (fondamentale per la stabilità dell'addestramento e per i bound di generalizzazione di Rademacher) è precisamente la condizione

che la rete sia un morfismo nella categoria arricchita con la metrica di Lipschitz.

IV.12 Funtori liberi, comonade della counità e i transformer come computazioni libere

La struttura dell'architettura transformer può essere analizzata attraverso il framework delle **comonadi** — duali delle monadi — che modellano computazioni con "contesto".

Una **comonade** (W, ϵ, δ) è la duale di una monade:

- $W: \mathcal{C} \rightarrow \mathcal{C}$ funtore
- $\epsilon: W \Rightarrow \text{Id}_{\mathcal{C}}$ la **counità** (o extract)
- $\delta: W \Rightarrow W^2$ la **comoltiplicazione** (o duplicate)

con leggi duali alle leggi monadali. La comonade modella "computazioni con accesso al contesto": dato un elemento $w \in W(A)$, $\epsilon_A(w)$ estrae il valore "focale" in A , e $\delta_A(w) \in W(W(A))$ distribuisce il contesto su tutte le posizioni.

Il meccanismo di attenzione self-attention è una comonade sull'insieme delle sequenze: ogni posizione della sequenza ha accesso all'intero contesto ($W = \text{"sequenza"}$), la counità estrae il token alla posizione corrente ($\epsilon = \text{proiezione sulla posizione } i$), e la comoltiplicazione distribuisce il contesto ad ogni posizione ($\delta = \text{replica della sequenza come contesto per ogni posizione}$).

Più precisamente, la comonade del **reader** (o environment): $W_E(A) = E \rightarrow A$ per un oggetto di contesto E , con:

$$\epsilon_A: (E \rightarrow A) \rightarrow A \text{ [eval all'elemento focale } e_0 \in E]$$

$$\delta_A: (E \rightarrow A) \rightarrow (E \rightarrow (E \rightarrow A)) \text{ [currying: } \delta_A(f)(e)(e') = f(e')]$$

Il self-attention transformer è il caso $E = \mathbb{R}^{n \times d}$ (l'intera sequenza come contesto), e la funzione di attenzione è un morfismo di comodule — un morfismo che commuta con la struttura di comonade, preservando l'accesso contestuale.

IV.13 La corrispondenza di Curry-Howard-Lambek

La corrispondenza di Curry-Howard (tipi = proposizioni, programmi = dimostrazioni) si estende a una tripletta grazie al contributo di Lambek (1980):

Tipi = Proposizioni = Oggetti in una CCC

Programmi = Dimostrazioni = Morfismi in una CCC

Riduzione = Trasformazione = Uguaglianza in una CCC

Il triangolo completo:

Logica (Curry-Howard): il tipo $A \rightarrow B$ corrisponde all'implicazione $A \supset B$; il termine $f: A \rightarrow B$ corrisponde alla dimostrazione di $A \supset B$; la β -riduzione $f(a)$ corrisponde all'eliminazione del modus ponens.

Teoria delle categorie (Lambek): i tipi sono oggetti in una CCC, i termini sono morfismi, la β -riduzione è l'equazione $\text{eval} \circ \langle \text{curry}(f), \text{id} \rangle = f$.

Informatica teorica (tipi): i tipi come $A \times B$ (prodotto), $A + B$ (coprodotto/ either), $A \rightarrow B$ (funzione) corrispondono rispettivamente ai connettivi logici $\wedge$, $\vee$, $\supset$ e agli oggetti categoriali $\times$ (prodotto), $+$ (coprodotto), B^A (esponentiale).

Nella versione per i tipi dipendenti (Martin-Löf 1975, Curry-Howard-Lambek esteso):

- La quantificazione universale $\forall x:A.B(x)$ corrisponde al tipo $\Pi(x:A)B(x)$ e all'oggetto categoriale $\Pi_{\{x:A\}}B(x)$ (il Π prodotto dipendente in un topos locale)
- La quantificazione esistenziale $\exists x:A.B(x)$ corrisponde a $\Sigma(x:A)B(x)$ e $\Sigma_{\{x:A\}}B(x)$ (il Σ coprodotto dipendente)
- L'uguaglianza proposizionale $a = b$ corrisponde al tipo identità $\text{Id}_A(a,b)$ e all'oggetto dei "percorsi" tra a e b nella HoTT (cfr. sezione D.7 della monografia)

La Homotopy Type Theory è precisamente la versione ∞ -categoriale di questa corrispondenza: i tipi sono spazi (∞ -groupoids), i programmi sono punti, le prove di uguaglianza sono percorsi, le prove di uguaglianza tra prove sono omotopie, ecc.

IV.14 Applicazioni alla semantica denotazionale dei linguaggi neurali

La semantica denotazionale dei linguaggi di programmazione (Scott e Strachey 1971) usa la teoria delle categorie per assegnare significati matematici ai programmi. Il collegamento con i modelli linguistici neurali è non-ovvio ma illuminante.

Domini di Scott: la semantica denotazionale usa **domini** — poset completi per ordine diretto con una mappa di approssimazione — come oggetti. Un tipo ricorsivo come $A = A \rightarrow A$ (funzione da A a A) viene interpretato come il limite di una successione di approssimazioni nel dominio. La soluzione all'equazione di punto fisso $X = F(X)$ (fondamentale per la ricorsione) è garantita dal teorema di Knaster-Tarski (cfr. Appendice VI, sezione VI.8).

Semantica di continuazione per i transformer: il processo di generazione autoregressiva — produrre un token alla volta condizionando tutti i precedenti — è naturalmente descritto come una computazione in CPS (Continuation-Passing Style). Ogni passo di generazione produce una "continuazione" che è la distribuzione condizionale sul token successivo. La sequenza generata è il punto fisso di un operatore di continuazione:

$$T: (\text{Voc}^* \rightarrow \text{Distr}(\text{Voc})) \rightarrow (\text{Voc}^* \rightarrow \text{Distr}(\text{Voc}))$$

$$T(f)(w_1, \dots, w) = \text{distribuzione di } w_{+1} \text{ condizionata a } w_1, \dots, w$$

Il modello linguistico addestrato è (un'approssimazione di) il punto fisso $T(f) = f$ nella categoria appropriata — la distribuzione stazionaria del processo autoregressivo che riproduce la distribuzione empirica del corpus.

Questo collegamento con la semantica denotazionale suggerisce che lo studio delle proprietà formali dei modelli linguistici potrebbe beneficiare delle tecniche della semantica dei domini: la composizionalità, la continuità, la monotonicità delle distribuzioni prodotte dai transformer sono proprietà che la teoria dei domini studia in modo rigoroso.

IV.15 Limite e colimite come costruzioni universali nei sistemi formali

I **limiti** e i **colimiti** sono le costruzioni universali fondamentali delle categorie, e appaiono in ogni parte della matematica come il modo "ottimale" di combinare strutture o estrarne parti.

Il **limite** di un diagramma $F: J \rightarrow \mathcal{C}$ è un oggetto $\lim F$ con morfismi compatibili verso ogni $F(j)$, universale nel senso che ogni cono

compatibile si fattorizza univocamente attraverso $\lim F$. Casi speciali: prodotto (J discreta), equalizzatore ($J = \bullet \rightarrow \bullet$), limite inverso ($J = \mathbb{N}^{\text{op}}$).

Il **colimite** è il duale: un oggetto $\text{colim } F$ con morfismi da ogni $F(j)$, universale per coconi compatibili. Casi speciali: coprodotto (J discreta), coequalizzatore, colimite diretto.

Esempi nella teoria dei sistemi formali:

L'algebra iniziale di un funtore $F: \mathcal{C} \rightarrow \mathcal{C}$ è un oggetto I con un morfismo $F(I) \rightarrow I$ che è un isomorfismo, e che è iniziale tra tutti gli F -algebre (oggetti A con morfismo $F(A) \rightarrow A$). Per $F(X) = 1 + X$ (l'aggiunta di un punto), l'algebra iniziale in Set è $\mathbb{N}$ (i numeri naturali) — il più piccolo tipo di dati ricorsivo. Il principio di induzione sui naturali è la proprietà universale dell'algebra iniziale.

La coalgebra finale (duale) di F è il sistema dinamico "massimale" coinvolto: per $F(X) = \Sigma \times X$ (prodotti con un alfabeto Σ), la coalgebra finale è Σ° — le sequenze infinite su Σ — il tipo dei flussi (streams) infiniti.

Connessione con le reti neurali: l'architettura residuale (ResNet) ha una struttura di coalgebra: ogni layer è un morfismo $h^{\{l+1\}} = h^{\{l\}} + f(h^{\{l\}})$, che corrisponde all'azione di una coalgebra per il funtore $F(X) = X$ (cioè, lo spazio di stati rimane lo stesso, e la mappa di transizione è il layer residuale). Il processo di addestramento cerca la coalgebra ottimale — quella il cui "comportamento" (la distribuzione prodotta) approssima meglio i dati di training.

Nota bibliografica dell'Appendice IV

Mac Lane, *Categories for the Working Mathematician*, seconda edizione (1978). Eilenberg e Mac Lane, *General Theory of Natural Equivalences*,

in Transactions of the American Mathematical Society, 58 (1945). Moggi, Notions of Computation and Monads, in Information and Computation, 93 (1991). Lambek, From λ -Calculus to Cartesian Closed Categories, in To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism (1980). Lawvere, Metric Spaces, Generalized Logic, and Closed Categories, in Rendiconti del Seminario Matematico e Fisico di Milano, 43 (1973). Kan, Adjoint Functors, in Transactions of the American Mathematical Society, 87 (1958). Johnstone, Sketches of an Elephant: A Topos Theory Compendium (2002). Fong e Spivak, An Invitation to Applied Category Theory: Seven Sketches in Compositionality (2019). Fong, Spivak e Tuyeras, Backprop as Functor: A Compositional Perspective on Supervised Learning, in Proceedings of LiCS (2019). Pearlmutter e Siskind, Reverse-Mode AD in a Functional Framework: Lambda the Ultimate Backpropagator, in ACM Transactions on Programming Languages and Systems, 30 (2008). Cohen e Welling, Group Equivariant Convolutional Networks, in Proceedings of ICML (2016). Cruttwell, Gavranović, Ghani, Wilson e Zanasi, Categorical Foundations of Gradient-Based Learning, in Proceedings of ESOP (2022). Scott e Strachey, Toward a Mathematical Semantics for Computer Languages, in Proceedings of the Symposium on Computers and Automata (1971). Martin-Löf, An Intuitionistic Theory of Types: Predicative Part, in Logic Colloquium (1975). Voevodsky e colleghi, Homotopy Type Theory: Univalent Foundations of Mathematics (2013). Aghajanyan, Gupta e Zettlemoyer, Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning, in Proceedings of ACL (2021).

Appendice V — Complessità algoritmica di Kolmogorov

V.1 Il problema della casualità e la sua misura

La teoria della complessità algoritmica nasce da una domanda apparentemente semplice: cosa significa che una stringa di simboli è *casuale*? La risposta intuitiva — che una stringa casuale non ha struttura, non ha pattern, non è comprimibile — viene resa precisa, in modo sorprendente, attraverso il concetto di calcolabilità.

L'intuizione di partenza si deve, in forma indipendente, a Ray Solomonoff (1960), Andrei Kolmogorov (1965) e Gregory Chaitin (1966). La formulazione oggi standard è quella di Kolmogorov. Si consideri una stringa binaria $x \in \{0,1\}^*$. La *complessità di Kolmogorov* di x , relativa a una macchina di Turing universale U , è:

$$K(x) = \min \{ |p| : U(p) = x \}$$

dove $|p|$ è la lunghezza in bit del programma p e il minimo è preso su tutti i programmi p che producono x come output (e si arrestano). In altre parole, $K(x)$ è la lunghezza del più breve programma che, eseguito su U , genera x . Se nessun programma genera x , la complessità non è definita (ma per stringhe finite questo caso non si presenta).

La definizione dipende dalla scelta della macchina universale U . Se U e U' sono due macchine universali distinte, le complessità corrispondenti K e K' differiscono di al più una costante additiva che dipende solo dalla coppia (U, U') , non dalla stringa x :

$$|K_U(x) - K_{U'}(x)| \leq c_{\{U, U'\}}$$

Questo fatto — il *teorema di invarianza* — è la prima proprietà fondamentale della complessità di Kolmogorov, e ne garantisce il carattere oggettivo (a meno di costanti).

Dimostrazione del teorema di invarianza. Sia U una macchina universale e sia M una qualunque macchina di Turing. Poiché U è universale, esiste un programma simulatore p_M di lunghezza fissa c_M tale che, per ogni programma q : $U(p_M q) = M(q)$, dove $p_M q$ denota la concatenazione. Dunque, per ogni stringa x , se $M(q) = x$, allora $U(p_M q) = x$, e $|p_M q| = c_M + |q|$. Prendendo il minimo su tutti i q tali che $M(q) = x$:

$$K_U(x) \leq K_M(x) + c_M$$

Scambiando i ruoli di U e M si ottiene il bound simmetrico, da cui la differenza è limitata. ■

Fissata una macchina universale di riferimento U (che d'ora in avanti supporremo fissa e ometteremo dall'indice), $K(x)$ misura la *descrizione minima* di x : la quantità di informazione necessaria e sufficiente per specificare x in modo univoco.

V.2 Proprietà elementari della complessità

Prima di dimostrare i risultati principali, è utile raccogliere alcune proprietà immediate che illustrano il comportamento di K .

Proposizione V.1 (Bound superiore triviale). Per ogni stringa x di lunghezza n :

$$K(x) \leq n + c$$

per una costante c indipendente da x . In altre parole, la complessità di una stringa non supera mai la sua lunghezza più una costante.

Dimostrazione. Il programma che consiste nella stringa x stessa, prefissato da un'istruzione di lettura e stampa di lunghezza fissa c , è un programma di lunghezza $n + c$ che produce x . Dunque $K(x) \leq n + c$. ■

Proposizione V.2 (Stringhe regolari). Se x è la stringa 0^n (n zeri consecutivi), allora $K(x) \leq \log_2(n) + c$. Più in generale, se x è descritta da una formula compatta (come la sequenza dei primi n decimali di π), allora $K(x)$ è molto più piccola di n .

Dimostrazione. Il programma "stampa n zeri" ha lunghezza proporzionale a $\log_2(n)$ (la lunghezza necessaria per codificare il numero n), più una costante per l'istruzione. ■

Proposizione V.3 (K non è calcolabile). La funzione $x \mapsto K(x)$ non è calcolabile: non esiste nessun programma che, dato x come input, si arresti sempre e restituisca $K(x)$.

Dimostrazione. Supponiamo per assurdo che K sia calcolabile. Si costruisce la seguente macchina M : dato n in input, M enumera tutte le stringhe x in ordine crescente di lunghezza, calcola $K(x)$ per ciascuna, e si arresta non appena trova la prima stringa x tale che $K(x) > n$. Sia x_n tale stringa. Il programma che descrive M e fornisce n come input ha lunghezza $O(\log n)$, ma M restituisce una stringa x_n con $K(x_n) > n$. Per n sufficientemente grande, $O(\log n) < n$, e si ottiene una contraddizione: x_n ha un programma di lunghezza $O(\log n) < K(x_n)$. Assurdo. ■

Questa indimostrabilità è una forma di incompletezza: per ogni sistema formale S che può ragionare sulla complessità di Kolmogorov, esistono stringhe la cui complessità non può essere dimostrata in S (si veda la sezione V.5 sul teorema di Chaitin).

V.3 L'incompressibilità della maggioranza delle stringhe

Il risultato centrale della teoria elementare è che *quasi tutte* le stringhe sono incompressibili: il loro programma minimo è lungo quanto la stringa stessa. Formalmente:

Teorema V.4 (Incompressibilità della maggioranza). Per ogni $n \geq 1$ e ogni $c \geq 0$, il numero di stringhe binarie di lunghezza n con $K(x) < n - c$ è al più $2^{\{n-c+1\}} - 1 < 2^{\{n-c+1\}}$. In particolare, la frazione di stringhe di lunghezza n con $K(x) < n - c$ è al più $2^{\{-c+1\}}$.

Dimostrazione. Il conteggio è l'unico strumento necessario. Consideriamo l'insieme di tutti i programmi p di lunghezza strettamente minore di $n - c$. La loro cardinalità totale è:

$$|\{p : |p| < n - c\}| = \sum_{k=0}^{n-c-1} 2^k = 2^{\{n-c\}} - 1$$

Ogni programma di lunghezza $k < n - c$ genera al più una stringa (perché i programmi per macchine con output deterministico generano al più un output). Dunque il numero di stringhe x con $K(x) < n - c$ è al più $2^{\{n-c\}} - 1$.

Il totale delle stringhe di lunghezza n è 2^n . La frazione comprimibile è quindi:

$$(2^{\{n-c\}} - 1) / 2^n < 2^{\{n-c\}} / 2^n = 2^{\{-c\}}$$

La frazione di stringhe *incompressibili* (con $K(x) \geq n - c$) è almeno $1 - 2^{\{-c\}}$. Per $c = 0$, almeno la metà delle stringhe di lunghezza n è incompressibile (nel senso che $K(x) \geq n$). Per $c = 1$, almeno tre quarti lo sono. Per $c \rightarrow \infty$, la frazione incompressibile tende a 1. ■

Corollario V.5. Per ogni n , esistono stringhe di lunghezza n con $K(x) \geq n$. Tali stringhe si dicono *algoritmicamente casuali* (o incompressibili nel senso forte).

Corollario V.6. La proprietà " $K(x) \geq n - c$ " è verificata dalla "quasi totalità" delle stringhe in un senso misurabile preciso: se si sceglie uniformemente a caso una stringa di lunghezza n , la probabilità che essa sia comprimibile di almeno c bit è inferiore a 2^{-c+1} .

Il teorema V.4 ha una natura puramente combinatoria: dipende soltanto dal fatto che ci sono più stringhe di lunghezza n che programmi di lunghezza minore di $n - c$. È un argomento di pigeonhole generalizzato. La computabilità non entra: il risultato vale per qualunque nozione di "descrizione" che soddisfi il principio che descrizioni diverse generano output diversi.

V.4 Stringhe incompressibili e sequenze normali

Si può approfondire la caratterizzazione delle stringhe incompressibili. Una stringa x di lunghezza n è *c-incompressibile* se $K(x) \geq n - c$. Le stringhe 0-incompressibili ($K(x) \geq n$) sono quelle per cui non esiste alcuna compressione. Esse soddisfano proprietà combinatoriche che le rendono indistinguibili dalle stringhe "casuali" in ogni test statistico calcolabile.

Proposizione V.7. Sia x una stringa binaria di lunghezza n con $K(x) \geq n - c$. Allora, per ogni $c' > c$, la stringa x non può essere generata da nessun programma di lunghezza $n - c'$.

Più importante è il legame con la *normalità* nel senso di Borel. Una stringa x di lunghezza n si dice *ε -normale* se, per ogni blocco binario w di lunghezza $k \leq \log_2(n) - \log_2(1/\varepsilon)$, la frequenza di w in x è compresa tra $2^{-k} - \varepsilon$ e $2^{-k} + \varepsilon$. Cioè ogni blocco compare con la frequenza che ci si aspetta per una scelta uniforme.

Teorema V.8. Ogni stringa c -incompressibile è $O(c/n)$ -normale.

Dimostrazione (schema). Supponiamo per assurdo che x di lunghezza n sia c -incomprimibile ma che qualche blocco w di lunghezza k compaia con frequenza anomala. Allora la distribuzione dei blocchi in x è diversa da quella uniforme, il che implica che x ha struttura statistica. La codifica aritmetica di x rispetto alla distribuzione osservata produce un programma di lunghezza approssimativamente $n \cdot H$ (distribuzione osservata), dove H è l'entropia di Shannon. Se la distribuzione è anomala, $H < 1$, e la lunghezza del programma è strettamente minore di $n - c$ (per c abbastanza piccolo rispetto alla deviazione statistica). Contraddizione con la c -incomprimibilità. ■

Questo teorema costituisce il ponte tra la complessità algoritmica e la teoria dell'informazione di Shannon: le stringhe incompressibili nel senso di Kolmogorov sono normali nel senso statistico, ed entrambe le caratterizzazioni convergono verso l'idea intuitiva di "sequenza casuale".

V.5 Il teorema di Chaitin sull'incompletezza

Il risultato più profondo della teoria della complessità algoritmica è il *teorema di Chaitin*, che fornisce un nuovo tipo di incompletezza, strutturalmente diverso da quello di Gödel ma complementare ad esso.

Teorema V.9 (Chaitin, 1971). Sia S un sistema formale coerente, sufficientemente potente da formalizzare la teoria della computazione. Esiste una costante c_S (dipendente da S) tale che S non può dimostrare affermazioni della forma " $K(x) > c_S$ " per nessuna stringa specifica x .

Più precisamente: per ogni costante $L > c_S$, l'insieme $\{ x : S \vdash "K(x) > L" \}$ è vuoto.

La formulazione equivalente: per ogni teorema dimostrabile in S della forma " $K(x) > m$ ", si ha $m \leq c_S$. Il sistema S non riesce a certificare complessità alta per nessuna stringa concreta.

Dimostrazione. La dimostrazione combina il principio di incompressibilità con l'auto-referenza, in uno spirito simile alla diagonalizzazione gödeliana.

Passo 1: costruzione del programma esploratore. Consideriamo il seguente algoritmo A , parametrizzato da una soglia L :

$A(L)$: enumera tutte le dimostrazioni di S in ordine canonico;

per ogni dimostrazione d trovata:

se d dimostra " $K(x) > L$ " per qualche stringa x :

return x

Poiché S è coerente e il suo insieme di teoremi è ricorsivamente enumerabile, l'algoritmo A è ben definito (anche se potrebbe non arrestarsi su tutti gli input). Se S dimostra " $K(x) > L$ " per qualche x , allora $A(L)$ troverà prima o poi tale dimostrazione e restituirà x .

Passo 2: stima della complessità di $A(L)$. Il programma che computa $A(L)$ ha lunghezza fissa c_S dipendente solo da S (contiene la descrizione dell'enumeratore di S), più la lunghezza della codifica di L . Per L di lunghezza k bit, il programma totale ha lunghezza $c_S + k \leq c_S + \log_2(L) + O(1)$.

Passo 3: la contraddizione. Supponiamo che S dimostri " $K(x) > L$ " per qualche stringa x . Allora $A(L)$ si arresta e restituisce x . Ma $A(L)$ è un programma per x di lunghezza $c_S + \log_2(L) + O(1)$. Quindi:

$$K(x) \leq c_S + \log_2(L) + O(1)$$

D'altra parte, S ha dimostrato $K(x) > L$. La contraddizione emerge quando:

$$L < K(x) \leq c_S + \log_2(L) + O(1)$$

Il membro destro cresce come $\log_2(L)$, mentre il sinistro cresce come L. Per ogni $L > c_S + \log_2(L) + O(1)$ — cioè per L abbastanza grande rispetto a c_S — si ha una contraddizione con la coerenza di S.

Passo 4: determinazione della costante. La costante critica c_S è il valore minimo di L tale che $L > c_S + \log_2(L) + O(1)$. Questo è un'equazione di punto fisso che ha sempre una soluzione finita (perché il membro sinistro cresce più rapidamente del destro). Per ogni L maggiore di tale valore, S non può dimostrare " $K(x) > L$ " per nessuna x. ■

Osservazione V.10. Il teorema di Chaitin è più forte dell'incompletezza gödeliana in un senso preciso: non si limita a esibire *una* proposizione indecidibile, ma mostra che *tutte* le proposizioni di una certa forma (le affermazioni di alta complessità) sono indimostrabili. Il sistema formale ha un "soffitto" algoritmico che non può superare.

Corollario V.11 (Incompletezza via complessità). Esistono stringhe x per cui $K(x) > c_S$ è vera ma non dimostrabile in S. Quindi i numeri di Kolmogorov definiscono una fonte sistematica di proposizioni vere e indimostrabili.

Osservazione V.12 (Il numero di Chaitin Ω). Il teorema di Chaitin è strettamente connesso al *numero* Ω (Chaitin, 1975), definito come la probabilità di arresto di una macchina universale prefisso-libera:

$$\Omega = \sum_{\{p \mid U(p) \text{ si arresta}\}} 2^{-|p|}$$

dove la somma è su tutti i programmi p per cui U si arresta, e il fattore $2^{-|p|}$ è la probabilità di generare p con lanci di moneta. Ω è un

numero reale tra 0 e 1 i cui bit codificano la soluzione del problema della fermata per ogni programma di lunghezza fissata. I bit di Ω sono "massimamente irriducibili": $K(\Omega \upharpoonright n) \geq n - c$ per ogni n (dove $\Omega \upharpoonright n$ denota i primi n bit di Ω), il che significa che Ω è algoritmicamente casuale nel senso più forte. Di conseguenza, per ogni sistema formale S , solo i primi c_S bit di Ω sono determinabili dai teoremi di S : il resto trascende irrimediabilmente la dimostrabilità formale.

V.6 Relazione con la compressione, l'entropia e l'apprendimento

La teoria di Kolmogorov è il fondamento teorico della compressione dati senza perdita. Il teorema di incompressibilità della maggioranza (V.4) implica immediatamente che nessun algoritmo di compressione lossless può comprimere *tutte* le stringhe: ogni algoritmo che comprime alcune stringhe ne allungherà necessariamente delle altre (argomento del pigeonhole). I formati come ZIP o LZ77 sfruttano la struttura *statistica* delle stringhe (la loro distribuzione non-uniforme nell'uso reale), non una proprietà universale.

Il legame con l'entropia di Shannon è il seguente. Se X è una variabile aleatoria con distribuzione p su un alfabeto finito, l'entropia di Shannon è $H(X) = -\sum p(x) \log_2(p(x))$. La complessità media di Kolmogorov di X — $E[K(X)]$ — soddisfa:

$$H(X) \leq E[K(X)] + c$$

per una costante c . In senso opposto, per distribuzioni computable, $E[K(X)] \leq H(X) + c'$. L'entropia di Shannon è dunque la media della complessità di Kolmogorov, ma la complessità di Kolmogorov misura

qualcosa di più fine: la complessità individuale di ogni stringa, non soltanto quella media.

Nel contesto dell'apprendimento automatico, la complessità di Kolmogorov formalizza il *rasoio di Occam* algoritmico: tra tutti i modelli compatibili con i dati osservati, il più semplice (quello con la più bassa complessità di Kolmogorov) è quello da preferire. Il *principio di minima descrizione* (MDL, Minimum Description Length) di Rissanen è una versione pratica e calcolabile di questo principio, in cui la complessità di Kolmogorov è sostituita dalla lunghezza della descrizione in un linguaggio di codifica fissato.

Nota bibliografica. I risultati originali si trovano in: A. Kolmogorov, "Three approaches to the quantitative definition of information", Problems of Information Transmission 1 (1965); G. Chaitin, "On the length of programs for computing finite binary sequences", Journal of the ACM 13 (1966); G. Chaitin, "A theory of program size formally identical to information theory", Journal of the ACM 22 (1975). Un'esposizione sistematica è in M. Li e P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer.

Appendice VI — Teoria degli ordini e dei reticoli nella logica

VI.1 Ordini parziali e strutture ordinate

La logica e la matematica intera sono percorse da strutture d'ordine. Un *ordine parziale* (o *poset*, da *partially ordered set*) è una coppia $\langle P, \leq \rangle$ dove $\leq$ è una relazione binaria su P che soddisfa:

- **Riflessività:** $x \leq x$ per ogni $x \in P$
- **Antisimmetria:** se $x \leq y$ e $y \leq x$, allora $x = y$
- **Transitività:** se $x \leq y$ e $y \leq z$, allora $x \leq z$

Un *ordine totale* (o *lineare*) aggiunge la condizione di *totalità*: per ogni $x, y \in P$, o $x \leq y$ oppure $y \leq x$. Esempi: $\langle \mathbb{N}, \leq \rangle$ è un ordine totale; $\langle \mathcal{P}(A), \subseteq \rangle$ (l'insieme delle parti di A ordinato per inclusione) è un ordine parziale non totale (per $|A| \geq 2$).

Un elemento $m \in P$ è un *massimale* se non esiste $x \in P$ con $m < x$ (ossia $m \leq x$ e $m \neq x$). Un *massimo* (o *top*) è un elemento $\top$ tale che $x \leq \top$ per ogni $x \in P$. Analogamente si definiscono *minimale* e *minimo* (o *bottom*, $\perp$). Non tutti i poset hanno massimo o minimo.

Dati $A \subseteq P$, un elemento $m \in P$ è un *upper bound* (o *maggiorante*) di A se $a \leq m$ per ogni $a \in A$. Il *least upper bound* (o *join*, o *estremo superiore*) di A , indicato $\sqcup A$ o $\bigvee A$, è il più piccolo dei maggioranti di A , se esiste. Simmetricamente, il *greatest lower bound* (o *meet*, o *estremo inferiore*) è il più grande dei minoranti.

VI.2 Reticoli: definizione e prime proprietà

Un *reticolo* è un poset $\langle L, \leq \rangle$ in cui ogni coppia di elementi $\{x, y\}$ ha sia un join che un meet:

$$x \vee y = \text{least upper bound di } \{x, y\}$$

$$x \wedge y = \text{greatest lower bound di } \{x, y\}$$

Equivalentemente, un reticolo può essere definito in modo puramente algebrico come una struttura $\langle L, \vee, \wedge \rangle$ in cui $\vee$ e $\wedge$ sono operazioni binarie che soddisfano, per ogni $x, y, z \in L$:

- **Idempotenza:** $x \vee x = x, x \wedge x = x$
- **Commutatività:** $x \vee y = y \vee x, x \wedge y = y \wedge x$
- **Associatività:** $(x \vee y) \vee z = x \vee (y \vee z)$, e analoga per $\wedge$
- **Assorbimento:** $x \vee (x \wedge y) = x, x \wedge (x \vee y) = x$

Le due definizioni sono equivalenti: da $\vee$ e $\wedge$ si ricava $\leq$ ponendo $x \leq y \leftrightarrow x \vee y = y$ (equivalentemente, $x \wedge y = x$); dalle proprietà di ordine si ricavano le operazioni.

Un reticolo è *limitato* (o *bounded*) se ha un minimo $\perp$ e un massimo $\top$. In un reticolo limitato, $\perp \vee x = x$ e $\top \wedge x = x$ per ogni x .

Un reticolo *distributivo* soddisfa la legge distributiva:

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

(Si può dimostrare che questa implica anche la legge duale $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.) I reticoli distributivi sono quelli in cui $\wedge$ e $\vee$ si comportano analogamente a $\cdot$ e $+$ nell'algebra.

Esempio VI.1. L'insieme delle parti $\mathcal{P}(A)$ con $\cap$ e $\cup$ è un reticolo distributivo limitato ($\perp = \emptyset, \top = A$). L'insieme dei divisori di un intero n con il massimo comun divisore ($\wedge$) e il minimo comune multiplo ($\vee$) è un reticolo distributivo. Il reticolo dei sottospazi vettoriali di uno spazio vettoriale è un reticolo *non* distributivo (in generale).

VI.3 Algebre di Boole

Un'algebra di Boole è un reticolo distributivo e complementato: a ogni elemento x è associato un *complemento* x' tale che:

$$x \vee x' = \top, x \wedge x' = \perp$$

e il complemento è unico (questo segue dalla distributività). Le algebre di Boole soddisfano le *leggi di De Morgan*:

$$(x \vee y)' = x' \wedge y'$$

$$(x \wedge y)' = x' \vee y'$$

Teorema VI.2 (Stone, 1936). Ogni algebra di Boole è isomorfa a un'algebra di insiemi: esiste uno spazio topologico X (lo *spazio di Stone*) tale che l'algebra è isomorfa alla collezione degli insiemi simultaneamente aperti e chiusi di X , con $\cup$ come $\vee$, $\cap$ come $\wedge$ e complemento come $'$.

Il collegamento con la logica classica è immediato: l'algebra degli enunciati del calcolo proposizionale classico, quozientata per equivalenza logica ($\varphi \equiv \psi \leftrightarrow \vdash \varphi \leftrightarrow \psi$), forma un'algebra di Boole. Questa è precisamente la *algebra di Lindenbaum-Tarski* della logica classica, di cui si parlerà nella sezione VI.5.

VI.4 Algebre di Heyting e la logica intuizionista

Un'*algebra di Heyting* è un reticolo distributivo e limitato in cui, per ogni coppia a, b , esiste il *relativo pseudo-complemento*: il massimo elemento c tale che $a \wedge c \leq b$. Questo elemento è indicato $a \rightarrow b$:

$$a \rightarrow b = \max \{ c \in L : a \wedge c \leq b \}$$

L'implicazione $a \rightarrow b$ è caratterizzata dalla legge di residuazione (o *adjunzione*):

$$c \leq a \rightarrow b \leftrightarrow a \wedge c \leq b$$

Questa legge è l'analogo algebrico della regola di deduzione: $\varphi, \psi \vdash \chi$ se e solo se $\varphi \vdash \psi \rightarrow \chi$.

Il complemento "booleano" può essere definito come $\neg a = a \rightarrow \perp$. In un'algebra di Heyting, $\neg$ è un operatore di complemento debole: vale $a \leq \neg \neg a$ (doppia negazione abbassa) ma non necessariamente $\neg \neg a \leq a$. La legge del terzo escluso $a \vee \neg a = \top$ non vale in generale. Un'algebra di Heyting in cui vale $a \vee \neg a = \top$ per ogni a è esattamente un'algebra di Boole.

Esempio VI.3. Sia X uno spazio topologico. L'insieme degli aperti di X , con $\cap$ come $\wedge$, $\cup$ come $\vee$ e la topologia come struttura d'ordine ($\subseteq$), è un'algebra di Heyting in cui:

$$U \rightarrow V = \text{int}(U^c \cup V)$$

dove int indica l'interno e U^c il complemento. Non è in generale un'algebra di Boole: il terzo escluso corrisponde alla proprietà che ogni aperto è anche chiuso (spazio discreto).

Le algebre di Heyting sono la semantica algebrica della logica intuizionista (si veda la sezione VI.6).

VI.5 La gerarchia di Lindenbaum

La costruzione di Lindenbaum-Tarski trasforma ogni logica in un'algebra. Sia L una logica proposizionale con un insieme For di formule. Si definisce la relazione di equivalenza:

$$\varphi \sim \psi \leftrightarrow \vdash \varphi \leftrightarrow \psi \text{ (}\varphi \text{ e } \psi \text{ sono logicamente equivalenti)}$$

Le classi di equivalenza $[\varphi]$ formano un insieme quoziente $\text{For}/\sim$, dotato di operazioni indotte dalle connettive logiche:

$$[\varphi] \vee [\psi] = [\varphi \vee \psi]$$

$$[\varphi] \wedge [\psi] = [\varphi \wedge \psi]$$

$$[\varphi] \rightarrow [\psi] = [\varphi \rightarrow \psi]$$

$$\neg[\varphi] = [\neg\varphi]$$

$$\perp = [\perp]$$

$$\top = [\top]$$

dove $\perp$ e $\top$ sono rispettivamente la formula contraddittoria e la tautologia. L'ordine è:

$$[\varphi] \leq [\psi] \leftrightarrow \vdash \varphi \rightarrow \psi$$

Teorema VI.4 (Lindenbaum-Tarski per la logica classica). La struttura $\langle \text{For}(\text{CPC})/\sim, \wedge, \vee, \neg, \perp, \top \rangle$ è un'algebra di Boole, detta *algebra di Lindenbaum* della logica classica CPC.

Dimostrazione (schema). Tutti gli assiomi delle algebre di Boole corrispondono a tautologie della logica classica. L'idempotenza corrisponde a $\vdash \varphi \vee \varphi \leftrightarrow \varphi$; la complementazione corrisponde a $\vdash \varphi \vee \neg\varphi$ (terzo escluso) e $\vdash \neg(\varphi \wedge \neg\varphi)$ (non contraddizione); la distributività è una tautologia verificabile per tavole di verità. ■

Teorema VI.5 (Lindenbaum-Tarski per la logica intuizionista). La struttura $\langle \text{For}(\text{IPC})/\sim, \wedge, \vee, \rightarrow, \perp \rangle$ è un'algebra di Heyting.

Il punto cruciale della costruzione di Lindenbaum è che non si tratta di una mera traduzione sintattica: il teorema di completezza afferma la coincidenza tra derivabilità sintattica e ordine algebrico. Per la logica classica:

$$\vdash \varphi \leftrightarrow [\varphi] = \top \text{ nell'algebra di Lindenbaum}$$

La gerarchia di Lindenbaum si generalizza ai sistemi modali, temporali e di ordine superiore. Per ogni teoria T (insieme di formule chiuso per deduzione), l'algebra di Lindenbaum di T è il quoziente $\text{For}/\sim_T$ dove φ

$\sim \vdash_T \psi \leftrightarrow \vdash_T \neg \psi$. Le teorie complete corrispondono ad algebre semplici (senza fattori diretti propri, o equivalentemente con al più due elementi per le algebre di Boole).

Corollario VI.6 (Corrispondenza teorie-filtri). Le teorie che estendono T corrispondono biunivocamente ai filtri dell'algebra di Lindenbaum di T . In particolare, le teorie complete massimali corrispondono agli *ultrafiltri* dell'algebra.

Questo corollario rivela la struttura della logica: lo spazio delle teorie massimali (complete) è precisamente lo spazio di Stone dell'algebra di Lindenbaum, e il teorema di Stone fornisce una rappresentazione concreta di questo spazio come spazio topologico.

VI.6 I modelli di Kripke e la semantica relazionale

Un *frame di Kripke* è una coppia $\langle W, R \rangle$ dove W è un insieme non vuoto di *mondi possibili* e $R \subseteq W \times W$ è una *relazione di accessibilità*. Un *modello di Kripke* è una tripla $\langle W, R, V \rangle$ dove $V : \text{Var} \rightarrow \mathcal{P}(W)$ è una *funzione di valutazione* che associa a ogni variabile proposizionale l'insieme dei mondi in cui è vera.

La soddisfacibilità di una formula φ in un mondo $w \in W$ (scritta $\langle W, R, V \rangle, w \models \varphi$, o più brevemente $M, w \models \varphi$) è definita ricorsivamente:

$$M, w \models p \leftrightarrow w \in V(p)$$

$$M, w \models \perp \text{ è sempre falso}$$

$$M, w \models \varphi \wedge \psi \leftrightarrow M, w \models \varphi \text{ e } M, w \models \psi$$

$$M, w \models \varphi \vee \psi \leftrightarrow M, w \models \varphi \text{ o } M, w \models \psi$$

$$M, w \models \varphi \rightarrow \psi \leftrightarrow M, w \not\models \varphi \text{ o } M, w \models \psi \text{ (classica)}$$

$M, w \models \Box\varphi \leftrightarrow$ per ogni v con wRv : $M, v \models \varphi$

$M, w \models \Diamond\varphi \leftrightarrow$ esiste v con wRv tale che $M, v \models \varphi$

Per la logica intuizionista, la clausola per l'implicazione cambia in modo sostanziale:

$M, w \models \varphi \rightarrow \psi \leftrightarrow$ per ogni v con $w \leq v$: se $M, v \models \varphi$ allora $M, v \models \psi$

dove la relazione di accessibilità è ora un *ordine parziale* $\leq$ (riflessivo e transitivo). Questa differenza cattura il carattere costruttivo dell'intuizionismo: $\varphi \rightarrow \psi$ vale in w non perché φ sia falso, ma perché in *ogni futuro* in cui diventa disponibile φ , diventa disponibile anche ψ .

Theorem VI.7 (Kripke, completezza per IPC). La logica intuizionista IPC è corretta e completa rispetto alla classe dei modelli di Kripke su frame con $\leq$ ordine parziale: $\vdash_{\text{IPC}} \varphi$ se e solo se φ è vera in ogni mondo di ogni modello di Kripke con ordine parziale.

Il teorema di Kripke è fondamentale perché fornisce alla logica intuizionista una semantica accessibile (non puramente algebrica) e dimostra che IPC non è la logica di nessun algebra di Boole (nessuna algebra a due valori ne è un modello), ma è invece la logica delle algebre di Heyting.

Proprietà della persistenza. Nei modelli di Kripke per IPC, la verità è *persistente*: se φ vale in w e $w \leq v$, allora φ vale in v . Questo rispecchia l'idea intuizionista che la conoscenza è monotona: una volta ottenuta, non si perde. Formalmente, per ogni formula φ e ogni coppia di mondi $w \leq v$:

$M, w \models \varphi \Rightarrow M, v \models \varphi$

Questo si dimostra per induzione strutturale sulla formula φ . La persistenza dipende crucialmente dalla clausola di Kripke per

l'implicazione: essa garantisce che $\varphi \rightarrow \psi$ persiste proprio perché ogni futuro accesso a φ produce ψ .

Logiche modali e condizioni sui frame. Per la logica modale classica, diverse logiche corrispondono a diverse proprietà della relazione di accessibilità R :

- **K**: nessuna condizione su R (assioma $\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$)
- **T**: R è riflessiva — $\Box\varphi \rightarrow \varphi$
- **K4**: R è transitiva — $\Box\varphi \rightarrow \Box\Box\varphi$
- **S4**: R è riflessiva e transitiva — $\Box\varphi \rightarrow \varphi$ e $\Box\varphi \rightarrow \Box\Box\varphi$
- **S5**: R è una relazione di equivalenza — $\varphi \rightarrow \Box\Diamond\varphi$

Questi sistemi formano una gerarchia di inclusione: $K \subseteq T \subseteq S4 \subseteq S5$. La corrispondenza tra assiomi e proprietà di frame è il contenuto fondamentale del *teorema di Sahlqvist*, che caratterizza una vasta classe di assiomi modali per le proprietà first-order dei loro frame corrispondenti.

VI.7 Semantica algebrica: la corrispondenza fondamentale

Il legame tra la semantica dei modelli di Kripke e la semantica algebrica è il contenuto del seguente teorema di rappresentazione:

Teorema VI.8 (Rappresentazione dei frame di Kripke come algebre di Heyting). Sia $\langle W, \leq \rangle$ un frame di Kripke per IPC (con $\leq$ ordine parziale). L'insieme degli *upset* (o *upper sets*) di W :

$$Up(W) = \{ U \subseteq W : \text{se } w \in U \text{ e } w \leq v, \text{ allora } v \in U \}$$

con le operazioni $\cap$, $\cup$ e $U \rightarrow V = \{ w \in W : \text{per ogni } v \geq w, \text{ se } v \in U \text{ allora } v \in V \}$ forma un'algebra di Heyting.

Dimostrazione. $\langle \text{Up}(W), \cap, \cup, \emptyset, W \rangle$ è un reticolo distributivo limitato: le intersezioni e le unioni di upset sono upset, $\emptyset$ e W sono upset. La verifica dell'implicazione: sia dato $w \in U \rightarrow V$ (come definita sopra). Se $w \leq v$, allora per ogni $u \geq v$ con $u \in U$ si ha $u \geq w$ e $u \in U$, dunque $u \in V$, e quindi $v \in U \rightarrow V$. Quindi $U \rightarrow V$ è un upset. La legge di residuazione $U \cap X \subseteq V \leftrightarrow X \subseteq U \rightarrow V$ segue direttamente dalla definizione. ■

Corollario VI.9. La funzione di valutazione V in un modello di Kripke per IPC può essere equiparata a un omomorfismo d'algebre di Heyting dall'algebra di Lindenbaum di IPC in $\text{Up}(W)$.

Questa corrispondenza è il nucleo della *semantica algebrica*, che generalizza sia la semantica modello-teoreica che quella di Kripke. Il programma della semantica algebrica, sviluppato da Blok e Pigozzi, Rasiowa e Sikorski, e Jónsson e Tarski, ha lo scopo di studiare le logiche attraverso le varietà algebriche che le rappresentano:

Logica	Semantica algebrica
Logica classica CPC	Algebre di Boole
Logica intuizionista IPC	Algebre di Heyting
Logica modale S4	Algebre di Boole con operatore di chiusura (Algebre di McKinsey-Tarski)
Logica modale S5	Algebre di Boole con operatore di proiezione
Logica di Łukasiewicz a 3 valori	MV-algebre

Teorema VI.10 (Completezza algebrica, Rasiowa-Sikorski). Una formula φ è un teorema di IPC se e solo se, per ogni algebra di Heyting A e ogni omomorfismo $h : \text{For} \rightarrow A$, si ha $h(\varphi) = \top_A$.

Questo teorema — il cui lemma tecnico fondamentale, il *lemma di Rasiowa-Sikorski*, afferma che ogni elemento non-massimo di un'algebra σ -completa può essere separato da filtri propri preservando join numerabili — è la versione algebrica del teorema di completezza di Gödel per la logica del primo ordine.

VI.8 Connessioni di Galois e dualità

Il collegamento tra ordini e logica si approfondisce attraverso la teoria delle *connessioni di Galois*. Dati due poset $\langle P, \leq \rangle$ e $\langle Q, \leq \rangle$, una *connessione di Galois* è una coppia di funzioni (f, g) con $f : P \rightarrow Q$ e $g : Q \rightarrow P$ tali che:

$$f(x) \leq y \leftrightarrow x \leq g(y)$$

Le connessioni di Galois sono onnipresenti: l'implicazione in un'algebra di Heyting è caratterizzata da $a \wedge - \vdash a \rightarrow -$ (aggiunto destro di $a \wedge -$); la negazione $\neg$ è caratterizzata da $- \wedge \neg \leq \perp$ (residuale di $\wedge$ rispetto a $\perp$); i quantificatori $\forall$ e $\exists$ della logica del primo ordine sono aggiunto sinistro e destro dell'operazione di sostituzione di variabili.

Dualità di Stone. La dualità di Stone (1936) stabilisce un'equivalenza di categorie tra algebre di Boole e spazi topologici totalmente disconnessi e compatti (spazi di Stone):

$$\text{Algebre di Boole} \cong^{\text{op}} \text{Spazi di Stone}$$

L'algebra corrisponde alla sua struttura di aperti, e gli ultrafiltri dell'algebra corrispondono ai punti dello spazio. Questa dualità è il prototipo di tutte le dualità algebriche in logica, tra cui la *dualità di Priestley* (algebre distributive $\leftrightarrow$ spazi di Priestley) e la *dualità di Esakia* (algebre di Heyting $\leftrightarrow$ spazi di Esakia), che corrispondono rispettivamente alle semantiche di Kripke per le logiche intermedie.

La dualità di Esakia afferma:

Algebre di Heyting $\cong^{\text{op}}$ Spazi di Esakia

dove uno *spazio di Esakia* è uno spazio di Priestley $\langle X, \leq, \tau \rangle$ in cui $\uparrow U = \{ x : \exists y \in U, y \leq x \}$ è aperto per ogni aperto U . Attraverso questa dualità, i frame di Kripke per IPC sono esattamente i fibrati delle algebre di Heyting, e il teorema di completezza di Kripke è un corollario del teorema di rappresentazione di Esakia.

Conclusione. La teoria degli ordini e dei reticoli non è un accessorio estetico della logica: è la sua struttura profonda. Ogni sistema logico ha la propria algebra caratteristica, ogni algebra è rappresentata geometricamente da uno spazio duale, ogni spazio duale è popolato dai "mondi" dei modelli di Kripke. La gerarchia di Lindenbaum, la semantica di Kripke e la semantica algebrica sono tre facce dello stesso oggetto matematico.

Nota bibliografica. I risultati fondamentali sulle algebre di Boole e la dualità di Stone si trovano in M.H. Stone, "The theory of representations for Boolean algebras", Transactions of the AMS 40 (1936). La semantica algebrica della logica intuizionista è sviluppata in H. Rasiowa e R. Sikorski, *The Mathematics of Metamathematics*, PWN (1963). I modelli di Kripke sono introdotti in S.A. Kripke, "Semantical analysis of intuitionistic logic I", in J.N. Crossley e M.A.E. Dummett (a cura di), *Formal Systems and Recursive Functions*, North-Holland (1965). La dualità di Esakia è in L. Esakia, *Heyting Algebras: Duality Theory*, Metsniereba (1985); trad. ingl. Springer (2019).